Escuela de ciencias de la Ingeniería
Ingeniería Civil en Computación

Segmentación en imágenes hiperespectrales

Felipe Nicolás Gómez Molina
Profesor guía: Rodrigo Verschae
Profesor coguía: Luis Cossio

Memoria para optar al título de Ingeniero Civil en Computación

Rancagua, Chile
Diciembre 2023

Dedicatoria

Esta memoria está dedicada a mi hija, María Fernanda, quién ha sido mi principal motivo de seguir adelante día a día.

Agradecimientos

A lo largo de mis años de estudio he recibido el apoyo y el cariño de una innumerable cantidad de personas, a las cuáles me gustaría retribuir dedicándole unas palabras.

Quiero iniciar agradeciendo a mi familia. Agradezco a mi hija, María Fernanda, agradezco a mi hermana, Camila Gómez, agradezco a mi madre, Mariluz Molina y agradezco a mi padre, Sergio Gómez por su completo apoyo y comprensión durante estos cinco años de carrera, ya que, sin ellos, no sería posible estar donde estoy.

Agradezco a mis abuelos, María Teresa Morales, Eva Galarce y Sergio Gómez por su infinito afecto, consejos y conversaciones en mis momentos más difíciles.

Agradezco también a Elena Morales, madre de mi hija y Mónica Farías, abuela de mi hija, quienes han sido un pilar fundamental en la crianza y cuidado de mi bebé.

Agradezco a mis amigos más cercanos Paola Abarca, Vicente González, Hernán Moreno y Camila Rapu, con quienes he compartido durante los últimos años y han sido un apoyo tremendo cada vez que lo he necesitado. Más aún en el marco del desarrollo de esta investigación. Además, muchas gracias a cada uno de mis compañeros de carrera y universidad con quienes he aprendido y crecido a lo largo de estos años.

Finalmente quiero agradecer a todos y cada uno de los profesores que me guiado, orientado y enseñado más de lo que nunca pude haber imaginado. Particularmente quiero agradecer a Carol Moraga e Ignacio Bugueño, quienes confiaron en mí como nadie lo había hecho nunca, brindándome oportunidades que me permitieron crecer y desarrollarme profesionalmente. También, agradezco a mi profesor guía, Rodrigo Verschae y mi profesor coguía, Luis Cossio, quiénes me guiaron durante los últimos meses para completar esta investigación.

ÍNDICE

Resumen

La presente investigación tiene como principal objetivo la exploración y utilización de diversas técnicas de segmentación semántica utilizando información hiperespectral, es decir, información que contempla no sólo el espectro visible, sino que, además, contempla el espectro infrarrojo.

Para lograr este objetivo se implementarán las técnicas de segmentación utilizando la distancia euclidiana como métrica de comparación de píxeles y el enfoque del algoritmo watershed. Además, se utilizarán técnicas de aprendizaje de máquina no supervisado con enfoque en agrupamiento, estas técnicas son HDBSCAN y k-means. Todo el desarrollo fue realizado utilizando el lenguaje de programación Python y la librería Sikit-learn.

Los resultados mostraron que el algoritmo de k-means funciona particularmente bien en contextos de agroindustria, donde al contar con diferencias del espectro infrarrojo más marcadas en los elementos de interés, el algoritmo se ve fuertemente beneficiado del uso de tecnologías hiperespectrales.

Se logró concluir que el uso de imágenes hiperespectrales presenta una clara ventaja respecto al uso de imágenes con canales RGB dada la información adicional entregada por el espectro infrarrojo, la cual permite una diferenciación mucho más precisa que la que ofrece únicamente el color.

Esta investigación muestra que es posible tener un enfoque de segmentación fiable en caso de no contar con un conjunto de datos previamente etiquetados para poder utilizar técnicas más tradicionales como lo son redes neuronales.

Palabras clave: aprendizaje de máquina, HDBSCAN, hiperespectro, k-means, segmentación

1. Introducción

La exponencial evolución de las tecnologías ha facilitado la creación de una inmensa cantidad de nuevas herramientas para un sinfín de áreas y disciplinas completamente diferentes. Una de estas herramientas son las imágenes hiperespectrales, cuyas aplicaciones abordan campos de carácter regional tales como los de la agronomía o la minería, por ejemplo. Las imágenes hiperespectrales, a diferencia de las imágenes tradicionales, tienen la capacidad de obtener información de una amplia gama de longitudes de onda, lo que permite obtener un nivel de información mucho más precisa y detallada.

Un problema clásico para enfrentar en el análisis de imágenes digitales es la segmentación de regiones o áreas de interés dentro de la imagen. Tradicionalmente se trabaja con información entregada por los canales RGB para lograr la identificación de estos segmentos, sin embargo, usando información hiperespectral podemos obtener mucha más información, lo que puede ser de bastante utilidad a la hora de encontrar ciertas características que permitan determinar las regiones que componen a la imagen.

Con este enfoque en cuenta, se dará foco al análisis de imágenes hiperespectrales mediante el desarrollo de algoritmos de segmentación utilizando técnicas de visión computacional y machine learning. En particular, se buscará generar un sistema de segmentación a través de técnicas de aprendizaje automático y aplicación de algoritmos tradicionales, utilizando la cámara “Pika L de Resonon” y principalmente el lenguaje de programación Python.

La investigación se compondrá de tres principales objetivos. En primer lugar, se abordará la visualización de las imágenes hiperespectrales en formato RGB, con el propósito de conseguir una primera aproximación de la información contenida en las imágenes, como zonas de interés o patrones y, además, poder acceder de una manera fácil a la información hiperespectral contenida en cada píxel. Luego, se abordará la segmentación de las regiones a identificar utilizando los datos hiperespectrales, en donde se hará uso de modelos de aprendizaje no supervisado y aplicación de algoritmos no lineales. Y por último, se realizará una etapa de cuantificación de las regiones identificadas en la segmentación, en donde se identificará la cantidad de regiones diferentes que se tienen en la imagen, así como otras métricas de interés.

Esta Tesis está dividida en ocho secciones, donde la sección 1 se plantea la introducción al problema a trabajar, los objetivos a grandes rasgos y la estructura del escrito. En la sección 2 se planteará la hipótesis a de la investigación desarrollada. Las secciones 3 y 4 corresponden a los objetivos generales y específicos respectivamente. La sección 5 planteará todos los conocimientos teóricos requeridos para la investigación, así como una revisión de la literatura más relevante para el desarrollo del trabajo realizado. La sección 6 definirá en detalle cada uno de los pasos realizados en pro de la replicabilidad de los resultados de la investigación. Posteriormente, en la sección 7 se presentarán cada uno de los resultados obtenidos de los pasos descritos en la sección 6. Y finalmente, en la sección 8 se presentarán las conclusiones obtenidas de los resultados.

2. Hipótesis

Mediante el uso de imágenes hiperespectrales y a través de la combinación de técnicas de visión computacional y aprendizaje de máquina, es posible lograr una segmentación más precisa, detallada y representativa de las regiones de interés en la imagen en comparación a los métodos tradicionales basados únicamente en canales RGB.

3. Objetivo General

Se busca principalmente la implementación de algoritmos de segmentación en imágenes hiperespectrales para el reconocimiento y cuantificación de diferentes materiales utilizando métodos tradicionales y métodos de aprendizaje automático no supervisado.

4. Objetivos específicos

- **Objetivo específico N°1; Visualización:** Se busca lograr recibir imágenes hiperespectrales, almacenar el hiperespectro de cada píxel y visualizar imágenes a partir de anchos de banda escogidos por el usuario. En este objetivo buscamos representar la imagen hiperespectral como imagen RGB (300 canales → 3 canales), visualizar la información hiperespectral asociada a un píxel y visualizar la información hiperespectral asociada a un segmento de pixeles.
- **Objetivo específico N°2; Segmentación:** Se busca lograr la identificación de regiones de píxeles vecinos con características similares y clasificar regiones. En este objetivo se busca utilizar y evaluar algoritmos basados en aplicación de filtros no lineales para la segmentación y se busca utilizar y evaluar modelos de aprendizaje no supervisados para la segmentación.
- **Objetivo específico N°3; Cuantificación:** Se busca lograr la cuantificación del número y la fracción de área de las regiones segmentadas. En este objetivo se busca determinar la cantidad de regiones segmentadas encontradas. Además, se busca determinar la fracción que utilizan las regiones dentro del total de la imagen.

4.1. Consideraciones y limitaciones

Dentro del desarrollo de esta investigación se tuvieron en cuenta algunos puntos que es importante mencionar para entender de mejor manera las decisiones tomadas al momento de elegir el camino a seguir.

4.1.1 Limitaciones de los conjuntos de datos para aprendizaje supervisado

Para realizar la segmentación en enfoques de aprendizaje supervisado utilizando técnicas como redes neuronales artificiales o máquinas de vectores de soporte, es esencial tener un conjunto de imágenes pre etiquetadas. Sin embargo, en el caso de imágenes hiperespectrales, obtener este tipo de datos etiquetados es particularmente difícil ya que requiere de un extenso trabajo que debe ser realizado de manera manual, lo que es inviable en el tiempo establecido. Teniendo en cuenta esta limitación, se eligió un enfoque de clustering no supervisado ya que este enfoque no depende de los datos etiquetados.

4.1.2. Limitación de capacidad de cómputo con imágenes hiperespectrales

Procesar la gran cantidad de datos hiperespectrales (300 datos por píxel) representa un desafío significativo en contexto de capacidad de cómputo. La alta dimensionalidad de las imágenes trabajadas inevitablemente tiene repercusiones considerables en los requisitos tanto de memoria RAM como de tiempo de procesamiento, haciendo inviable el análisis directo de esta gran cantidad de datos. Por esta razón se optó por la utilización de herramientas de reducción de dimensionalidad como Análisis de Componentes Principales (PCA) o autoencoders.

5. Marco teórico y revisión de literatura

En esta sección se abordarán los principios teóricos y definiciones técnicas requeridas para el entendimiento que son relevantes en el desarrollo de esta investigación. Además, se realizará una revisión de la literatura que permita conocer el estado del arte en la actualidad.

5.1. Imagen Digital

5.1.1. Definición y conceptos clave

Una imagen digital se puede definir como una matriz bidimensional que se compone de un número finito de elementos llamados píxeles. Cada píxel tiene una cantidad de valores discretos que pueden representar múltiple información, como intensidad, brillo, color, etc. Dependiendo de la manera en la que se elija interpretar estos valores que componen cada píxel.

A diferencia de las imágenes analógicas, como las fotografías tradicionales en papel, las imágenes digitales son espacialmente discretas, lo que significa que tenemos una cantidad discreta de elementos (píxeles) que componen una imagen, así como una cantidad discreta en el dominio de la amplitud (brillo y color).

5.1.2. Representación digital de una imagen

Es posible determinar la calidad (también llamada resolución o definición) de una imagen mediante la cantidad de píxeles que la componen en un sentido horizontal y vertical, por ejemplo, una imagen compuesta por 1280 x 720 píxeles, es considerada como una imagen de alta definición (HD), una imagen 1920 x 1080 es considerada como de muy alta definición (Full HD).

La digitalización ha revolucionado múltiples y diversos campos, como lo son el arte con la fotografía y el cine, la medicina con las imágenes radiológicas digitales, la astronomía en la teledetección, entre muchas otras aplicaciones. Esto se debe a que las imágenes digitales ofrecen un almacenamiento mucho más eficiente que su contraparte analógica, además de una manipulación mucho más precisa y una transmisión casi instantánea en algunos casos con las tecnologías actuales.

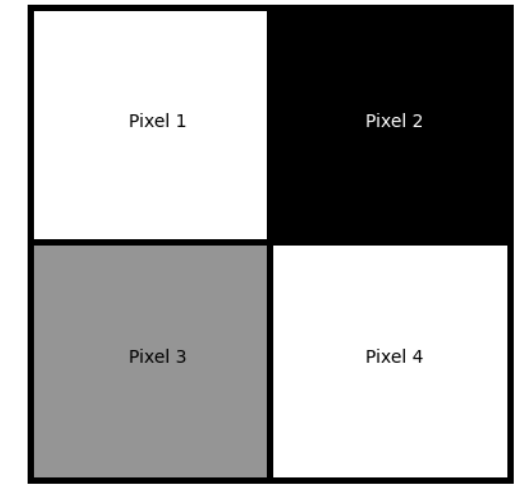

Figura 1. Imagen de 2x2 píxeles.

5.1.3. Visualización de una imagen digital

Como se mencionó previamente, cada píxel tiene una cantidad discreta de valores asociados, los que típicamente están dados entre un rango de 0 a 1 o 0 a 255. De esta manera, se puede obtener una representación numérica de un color. Cada uno de estos valores dentro de un píxel, son denominados como "canal".

La manera en la que los valores numéricos dentro del píxel son interpretados depende del espacio de color que se esté utilizando. Algunos de los espacios de color más comúnmente utilizados son: escala de grises, BGR o RGB, HSV, entre otros.

La manera más simple y básica de visualizar una imagen digital, es la escala de grises, que corresponde a un espacio de color en el que cada píxel contiene únicamente un sólo valor o canal, que es representado con 0 para el color blanco, 1 o 255 para negro y cualquier otro valor intermedio para representar diferentes tonalidades de gris. En el caso de la figura 1, por ejemplo, tanto el valor del píxel 1 como el valor del píxel 4 serían de 0, el color del píxel 2 sería de 1 o 255 y el valor del píxel 3 sería de 0.5 o 128 si se estuviese utilizando la escala de grises.

Además de la escala de grises, existen otras maneras de representar un píxel. La más comúnmente utilizada es el modelo de color RGB, en donde, mediante tres canales se determina la intensidad presente en el píxel para el color rojo, el color verde y el color azul. Mediante la

combinación de estos tres colores y variando sus intensidades, es posible expresar cualquier color presente dentro del espectro visible.

También existen otro tipo de representaciones conocidas como imágenes multiespectrales, las cuales tienen un mayor número de canales que pueden representar información tanto del espectro visible como del espectro infrarrojo o ultravioleta.

Finalmente, existe un tipo de imágenes conocidas como hiperespectrales. Las imágenes hiperespectrales se caracterizan por tener una gran cantidad de canales, gracias a esto, es posible tener una suerte de continuidad en los canales que representa cada píxel de la imagen (ver Fig. 2). Este tipo de representación se consigue con cámaras especializadas para captar más información además de la entregada por el espectro visible. Gracias a esto, se obtiene una representación mucho más precisa de la realidad.

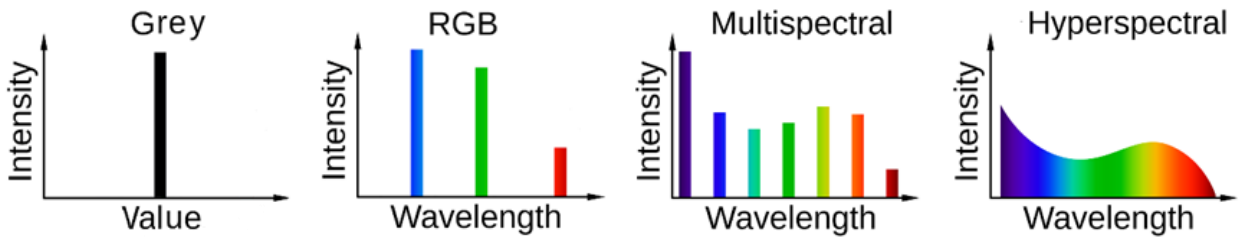

Figura 2. Comparación de escala de grises, RGB, multiespectro e hiperespectro. (Lucasbosch, 2021)

5.2. Visión computacional

La visión computacional es un campo de la informática cuyo objetivo es dotar a los computadores de visión y poder analizar datos como imágenes o videos en tiempo real de manera análoga a como lo haría un humano. Su enfoque principal es la adquisición, procesamiento y análisis de imágenes y videos digitales para extraer información de utilidad.

En la visión computacional se utilizan una gran cantidad de algoritmos y diversas técnicas para el procesamiento de todo tipo de imágenes digitales. Se desempeñan tareas como la detección de objetos, la segmentación de imágenes, el seguimiento de objetos, la reconstrucción tridimensional, la clasificación de imágenes, el reconocimiento de patrones, entre otros. Este

campo es ampliamente utilizado en áreas como la robótica, la medicina, automatización industrial, la astronomía, la realidad aumentada o realidad virtual, la agronomía, entre muchas más.

La visión computacional además requiere de tecnologías de hardware como lo son cámaras digitales, sensores de imágenes, sensores lidar, cámaras hiperespectrales u otro tipo de hardware especializado.

5.3. Segmentación

La segmentación es uno de los problemas más comunes del campo de la visión computacional. Se refiere a la división de regiones en base a similitud de características relacionadas al color, textura o intensidad. El objetivo es lograr marcar estos píxeles similares contiguos que representan áreas de la imagen. En la figura 3 se observa un ejemplo de segmentación.

Figura 3. Ejemplo de segmentación. (Artacho & Savakis, 2019)

A diferencia de otro problema clásico como lo es la detección de objetos, en la segmentación no interesa determinar qué elemento está presente en la imagen, sino más bien si hay elementos de un mismo tipo en la imagen, sin importar qué son.

5.3.1. Distancia Euclidiana

La distancia euclidiana, también conocida como norma euclidiana, es una medida que permite determinar cuál es la distancia espacial entre dos puntos. La distancia euclidiana entre dos puntos $A = (a_1, a_2, \ldots, a_n)$ y $B=(b_1, b_2, \ldots, b_n)$ está dado por:

$$D_E(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

Esta medida es usualmente utilizada para determinar la distancia más corta entre dos puntos en un plano. En el contexto de segmentación de imágenes, podemos utilizar el criterio de distancia como criterio de similitud entre dos puntos.

En una imagen compuesta por píxeles se debe seleccionar un píxel representativo del segmento buscado para así determinar mediante la distancia euclidiana si cada uno del resto de píxeles que componen la imagen pertenecen o no al segmento del píxel representativo.

5.3.2. Distancia de Mahalanobis

La distancia de Mahalanobis es una medida ampliamente utilizada para determinar la similitud entre un conjunto de valores desconocidos y un conjunto de valores conocidos. Esta distancia se diferencia de la distancia euclidiana porque tiene en cuenta la correlación entre variables de los datos. Por lo tanto, se considera más preciso y confiable cuando se trata de datos multidimensionales donde las variables pueden estar correlacionadas. Básicamente, la distancia de Mahalanobis mide la distancia entre un punto y una distribución, en lugar de la distancia entre dos puntos diferentes. Esto es particularmente útil en aplicaciones de detección y clasificación de anomalías, donde es importante medir cuánto se desvían los puntos de un conjunto de referencia.

La distancia de Mahalanobis está definida con bajo la siguiente fórmula.

$$D_{M(x)} = \sqrt{(y - \mu)^T \sum^{-1} (y - \mu)}$$

Donde D_M es la distancia de Mahalanobis, y es el vector de valores, μ es el vector de medias del conjunto de datos de referencia, $\sum^{-1}$ es la inversa de la matriz de covarianza del conjunto de datos de referencia y el símbolo ^{T} representa la matriz traspuesta.

5.3.3. Watershed

El algoritmo Watershed es un método popular de segmentación de imágenes que funciona bien para separar objetos en contacto o superpuestos en una imagen. El algoritmo se basa en la analogía de un paisaje o superficie de relieve, donde se visualiza la imagen como un mapa topográfico. La altura está representada por los valores de intensidad de los píxeles en este mapa. El algoritmo Watershed inunda este paisaje desde sus puntos más bajos (mínimos locales), lo que permite que se acumule "agua" y crezcan "cuencas". Una barrera se crea cuando el agua de diferentes cuencas está a punto de mezclarse, lo que define los límites de segmentación de la imagen. Esto conduce a la división de la imagen en regiones o segmentos distintos, cada uno correspondiendo a un objeto o parte de la imagen.

El algoritmo Watershed puede ser susceptible a la sobre segmentación, especialmente en imágenes con variaciones de intensidad o ruido, a pesar de su eficacia en la separación de objetos adyacentes. Para contrarrestar esto, con frecuencia se realiza un preprocesamiento de la imagen, como el uso de marcadores, filtros de suavizado o eliminación de ruido. Los marcadores son puntos o áreas seleccionadas manual o automáticamente dentro y fuera del objeto (marcadores internos) para guiar la inundación del Watershed y ayudar a controlar la segmentación. El algoritmo evita segmentar áreas innecesarias al usar marcadores para concentrarse en áreas específicas.

5.4. Aprendizaje de máquina

El aprendizaje de máquina, también conocido como aprendizaje automático o machine learning, es un área de la inteligencia artificial, en donde mediante algoritmos y modelos matemáticos, se busca dotar a las computadoras con un cierto grado de inteligencia. En lugar de

que las máquinas sean programadas para realizar una tarea específica, estas máquinas son capaces de determinar el proceso para realizar una tarea en base al análisis de resultados previamente obtenidos.

El objetivo primordial del aprendizaje automático es posibilitar que las computadoras adquieran conocimiento de forma automática, sin intervención ni ayuda humana, y que adapten su comportamiento en función de ello. Por ejemplo, un sistema de aprendizaje de máquina podría lograr discernir entre imágenes perteneciente o no a un animal específico.

Generalizando, existen principalmente dos tipos de aprendizaje de máquina, el aprendizaje supervisado y el aprendizaje no supervisado, donde la principal diferencia está en que en el enfoque supervisado se requiere de tener los datos previamente etiquetados, mientras que, en el enfoque no supervisado, se realiza sin etiquetas, permitiendo al modelo encontrar patrones o estructuras en los datos.

5.4.1. Aprendizaje no supervisado

El aprendizaje no supervisado es un método de aprendizaje automático donde se ajusta un modelo en base a la observación y reconocimiento de patrones en los datos. El enfoque más clásico es conocido como “clustering”, y se centra en agrupar datos espacialmente cercanos dentro de clusters o grupos.

5.4.1.1. k-means.

El algoritmo k-means es uno de los enfoques más comunes y utilizados del aprendizaje no supervisado, su enfoque principal es el agrupamiento de datos o clustering. Su objetivo es dividir un conjunto de datos en k subconjuntos o clusters, de tal manera que se consigan grupos con los datos de un grado de similaridad lo más alto posible, es decir, grupos que se compongan por datos lo más similares posibles entre ellos.

El aspecto más importante para considerar dentro de la utilización de k-means es el número de subconjuntos k que serán utilizados, ya que tiene un impacto significativo tanto en los resultados obtenidos como en la complejidad temporal de la ejecución del algoritmo.

Existen algunos enfoques para guiar la elección del número de k clusters óptimo, como por ejemplo el método del codo o el método de la silueta, sin embargo, estos enfoques requieren de la ejecución de diversos k para realizar la comparación y así determinar el número óptimo de k.

k-means a pesar de su simplicidad y eficiencia tiene la principal limitación de tender a funcionar mejor en clusters con forma esférica y tamaños similares, por lo que pude ser menos efectivo al momento de encontrar clusters con formas más irregulares o con datos que contienen una gran cantidad de ruido.

Además, es importante tener en cuenta que el algoritmo depende en gran medida de la posición inicial de los centroides, lo que puede llevar a conseguir diferentes resultados para dos ejecuciones de la misma configuración de algoritmo. Los dos principales tipos de inicialización de centroides son de manera aleatoria y k-means++, el cuál selecciona de una manera más inteligente y precisa las posiciones iniciales de los centroides. En el método K-means++, la selección ponderada de centroides busca eludir el impacto de una inicialización no óptima al asignar una probabilidad más alta a los puntos que están más lejos de los centroides ya seleccionados.

El algoritmo puede ser definido de manera simplificada en pseudocódigo de la siguiente manera:

Algoritmo: k-means
Entrada: Conjunto de n puntos D_1, D_2, D_n, número de clusters k.
Salida: Clusters C_1, C_2, ... C_k.
Paso 1: Posicionar k centroides en
coordenadas aleatorias.
Mientras no se cumpla el criterio de convergencia:
Paso 2: Para cada punto D_i se asigna un
cluster cuyo centroide sea el más cercano.
Paso 3: Para cada cluster C_i posicionar su centroide
al promedio de cada punto perteneciente a C_i.

Para la óptima selección de la cantidad k de clusters existen diversas técnicas, sin embargo, la más común se conoce como “método del codo”, el cual consiste en la evaluación de una métrica de clustering llamada “inercia”, al incrementar de uno en uno el número de clústeres. Al representar gráficamente el número de clusters y la inercia se puede observar un patrón que

mantiene una forma de codo y es aquí donde se selecciona la cantidad de cluster donde se desacelera de manera brusca.

La Inercia está definida de la siguiente manera:

$$I = \sum_{k=1}^{K} \sum_{i=1}^{n_k} ||x_i - c_k||^2$$

Donde I es la inercia, k es el número de clusters, n_k es el número de puntos en el cluster k, x_i es el i-ésimo punto en el cluster k y c_k es el centroide del cluster k.

5.4.1.2. DBSCAN.

Density-Based Spatial Clustering of Applications with Noise o DBSCAN es un algoritmo de clustering que se basa en la densidad espacial de los datos para formar grupos o clusters. A diferencia de k-means, DBSCAN no requiere que se especifiquen con anterioridad el número de clusters deseados. En cambio, identifica regiones de alta densidad que están separadas por regiones de baja densidad. Los puntos que se encuentren en las regiones de alta densidad serán considerados como parte del mismo grupo de datos, mientras que los puntos que se encuentren en áreas de baja densidad se consideran como ruido.

DBSCAN es útil para encontrar agrupaciones de diversas formas, a diferencia de k-means que encuentra clusters de forma esférica, por esta razón DBSCAN es mucho más versátil y útil para una amplia gama de aplicaciones.

DBSCAN funciona utilizando dos parámetros que deben ser seleccionados de manera manual. Épsilon (ϵ), el cuál es un parámetro que determina el radio a evaluar de un punto y MinPts que determina la cantidad mínima de puntos para considerar una región como "de alta densidad".

El algoritmo puede ser definido de manera simplificada en pseudocódigo de la siguiente manera:

```
Algoritmo: DBSCAN
Entrada: Conjunto de datos X, radio ϵ, umbral de densidad
         MinPts.
Salida: Clusters C₁, C₂, ..., Cₖ
Paso 1: Etiquetar todos los puntos en X como núcleo,
```

```
        frontera o ruido, basándose en el número de
        vecinos dentro del radio ϵ y el umbral MinPts.
Paso 2: Eliminar todos los puntos etiquetados como ruido.
Paso 3: Conectar todos los puntos núcleo que estén a una
        distancia menor o igual a ϵ entre sí.
Paso 4: Formar un clúster a partir de cada grupo de puntos
        núcleos conectados.
Paso 5: Asignar cada punto frontera al clúster más cercano
        de su punto núcleo.
```

5.4.1.3. HDBSCAN.

Hierarchical Density-Based Spatial Clustering of Applications with Noise o HDBSCAN es un algoritmo de agrupamiento avanzado derivado de DBSCAN. Su principal ventaja está en el análisis de clusters con alta densidad y su habilidad para manejar la aparición de ruido. La principal diferencia de HDBSCAN con DBSCAN está en su enfoque jerárquico, que le permite identificar clusters a diferentes escalas de densidad. Esta característica lo vuelve particularmente útil en conjuntos de datos donde se tiene una gran variabilidad de los datos en términos de densidad. A diferencia de k-means, HDBSCAN no necesita de un indicador de cuántos clusters se desea encontrar.

El algoritmo HDBSCAN comienza creando un árbol de jerarquía de clusters basado en la distancia y la densidad. Luego el árbol se simplifica a una versión más contenida y manejable, lo que se conoce como "árbol de cluster condensado" (McInnes & Healy, 2017).

El algoritmo puede ser definido de manera simplificada en pseudocódigo de la siguiente manera:

```
Algoritmo: HDBSCAN
Entrada: Conjunto de datos X, parámetro de tamaño mínimo de clúster
MinClusterSize.
Salida: Clústers C₁, C₂, ..., Cₖ.
Paso 1: Construir un árbol de expansión mínima a partir de X.
Paso 2: Crear un dendrograma a partir del árbol.
Paso 3: Simplificar el dendrograma en un árbol condensado.
Paso 4: Seleccionar clústers del árbol condensado basándose en
        MinClusterSize.
Paso 5: Asignar puntos a los clústers o como ruido.
```

5.5. Reducción de dimensionalidad.

La técnica de reducción de dimensionalidad es una herramienta clave para el aprendizaje de máquinas. Su importancia está en lograr reducir la complejidad de un set de datos, pero manteniendo la representatividad de estos, reduciendo así, los costes computacionales al momento de procesar los datos, pero obteniendo resultados equivalentes a trabajar con el set de datos original. Esta técnica es fundamental sobre todo en contextos de datos de una muy alta dimensionalidad, como sucede en el caso de trabajar con las imágenes hiperespectrales del set de datos utilizados en la presente investigación, donde cada píxel tiene una dimensión de 300.

Reducir el número de variables con las que se está trabajando no sólo simplifica el análisis, si no que, en algunos casos particulares, puede ayudar a prevenir el sobre-ajuste de los modelos.

Además de los beneficios técnicos que nos entrega la reducción de dimensionalidad, también nos ofrece la posibilidad de entender de mejor manera los datos a trabajar, ya que al reducir la dimensionalidad es posible observar a primera vista de mejor manera los patrones, agrupamientos o anomalías dentro del conjunto de los datos mediante la utilización de gráficos.

Los métodos más utilizados para realizar reducciones de dimensionalidad son el análisis de componentes principales (PCA) y los métodos basados en autoencoders en aprendizaje profundo.

5.5.1. PCA

El análisis de componentes principales o PCA es una técnica de reducción de dimensionalidad ampliamente utilizada en el procesamiento y análisis de datos. El análisis de componentes principales transforma el conjunto de variables que pueden o no estar correlacionadas en un conjunto de valores linealmente no correlacionados llamadas componentes principales.

Este proceso funciona basado en priorizar las componentes con mayor covarianza, por lo cual, se mantiene la mayor parte de la variabilidad en los datos conservando la menor cantidad de componentes.

El análisis de componentes principales es muy frecuentemente utilizado en el campo del aprendizaje automático, dado que permite mejorar la eficiencia de los algoritmos en términos de tiempo de ejecución.

El algoritmo puede ser definido de manera simplificada en pseudocódigo de la siguiente manera:

```
Algoritmo: PCA.
Entrada: Conjunto de datos X. Cantidad de vectores propios K
Salida: Componentes principales.
Paso 1: Normalizar los datos en X.
Paso 2: Calcular la matriz de covarianza de X.
Paso 3: Calcular vectores y valores propios de la
        matriz de covarianza.
Paso 4: Ordenar los vectores propios por sus valores propios
        en orden descendente.
Paso 5: Seleccionar los primeros K vectores propios.
Paso 6: Transformar X utilizando los K vectores propios
        seleccionados.
```

5.5.2. Autoencoders

Los autoencoders son un tipo de red neuronal artificial utilizada en diversos campos para conseguir la reducción de dimensionalidad y el aprendizaje de representaciones eficientes de datos. Utilizan un enfoque de aprendizaje no supervisado y están diseñados para codificar o comprimir su entrada y decodificar o descomprimir a su salida, pasando los datos a través de una serie de capas que progresivamente reducen su dimensión y luego las reconstruyen.

Además, a diferencia del análisis de componentes principales, los autoencoders son capaces de detectar y aprender relaciones no lineales complejas, lo que los vuelve más versátiles y poderosos para muchos más tipos de conjuntos de datos.

La arquitectura clásica de un autoencoder, ilustrada en la Figura 4, se caracteriza por un diseño de 'cuello de botella'. La primera fase del autoencoder, conocida como el codificador (encoder), se enfoca en reducir la dimensionalidad de los datos. Esto se logra a través de una serie de capas que progresivamente comprimen la información, conduciendo a una representación de menor densidad en la capa del cuello de botella, también conocida como 'bottleneck'. En esta etapa intermedia, es posible integrar capas adicionales para realizar otras funciones, con el objetivo de aprovechar los datos en su forma de dimensión reducida.

Posteriormente, en la fase final, se encuentran las capas del decodificador (decoder), que trabajan para reconstruir los datos, devolviéndolos a su dimensionalidad original. Esta reconstrucción busca ser lo más fiel posible a los datos iniciales, a pesar de la compresión previa.

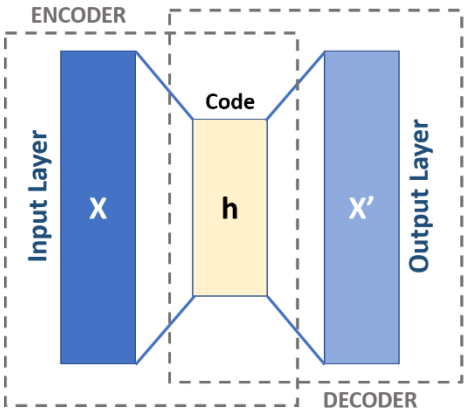

Figura 4. Arquitectura básica de un autoencoder.
(Massi, 2019).

5.6. Revisión de literatura

En este apartado se mencionarán algunos trabajos de investigación que entregan diversos enfoques que se relación con la segmentación en imágenes. Cada una de estas investigaciones fueron de gran utilidad para entender de mejor manera el problema abordad en esta investigación.

5.6.1. A Robust Stochastic Approach to Mineral Hyperspectral Analysis for Geometallurgy

Este documento, publicado en el año 2020, propone un innovador acercamiento al campo de la segmentación en imágenes hiperespectrales en el contexto de la geo metalurgia. Este documento tiene tres principales enfoques, primero, propone un nuevo algoritmo de segmentación en imágenes hiperespectrales, segundo, utiliza un esquema de reducción de dimensionalidad que preserva la información y tercero, explora un modelo de regresión jerárquica estocástica. Estos elementos entregan un marco muy amplio y robusto para el análisis de minerales de manera precisa y donde, además, se tienen en cuenta las variaciones de las condiciones ambientales y del contexto geológico.

El enfoque propuesto de este artículo es muy valioso en su campo porque mejora de manera significativa la capacidad de estimar variables críticas de contextos geológicos. Para el área de segmentación presenta aportes notables, dados sus enfoques de reducción de dimensionalidad y segmentación de imágenes.

5.6.2. Segment Anything Model

El Segment Anything Model o SAM, publicado el presente año, es un modelo innovador para el campo de la segmentación, debido a las múltiples maneras en las que puede recibir las instrucciones o "prompts", de esta manera le permite generar máscaras de información desde cajas de contorno, selección por puntos, máscaras o incluso texto. Además, el modelo está entrenado para reconocer una gran variedad de objetos y contextos.

Si bien este trabajo no utiliza imágenes hiperespectrales, de todos modos, entrega una visión muy valiosa respecto a la segmentación de imágenes. Introduce un enfoque completamente innovador respecto a la flexibilidad de la segmentación, además de los tratamientos para la ambigüedad en los "prompts" o inputs.

Además, este trabajo demuestra lo potente que puede ser un enfoque basado en aprendizaje supervisado, en particular, entrenando redes neuronales artificiales para el problema de la segmentación de imágenes y reconocimiento de objetos.

5.6.3. Hyperspectral image classification: A k-means clustering based approach

El artículo, publicado en el año 2017, nos entrega una metodología sencilla para enfrentar el problema de la segmentación de imágenes desde una aproximación de clustering. La metodología se basa en tres principales pasos, primero, se aplica el análisis de componentes principales para reducir la dimensionalidad de los datos, segundo, se aplica el algoritmo de clustering k-means para realizar un agrupamiento de los píxeles similares y conseguir una segmentación de la imagen y tercero, se aplica un algoritmo llamado máquina de vector de soporte multiclase (M-SVM) para entregar mayor precisión y robustez a las regiones encontradas.

Este método se validó con tres imágenes de referencia, en donde se demuestran mejoas en cuanto a precisión y tiempos de ejecución en comparación con la técnica estándar que sólo

contempla la aplicación del algoritmo de análisis de componentes principales y el algoritmo de la máquina de vector de soporte multiclase.

Este artículo fue de gran utilidad para la presente investigación, ya que, permitió conocer el estado del arte respecto a dos de los principales algoritmos que fueron utilizados durante el desarrollo, el análisis de componentes principales y k-means.

6. Marco metodológico

En esta sección se abordará el proceso paso a paso detallado realizado durante el desarrollo de la investigación. Se abordarán procedimientos y técnicas empleadas a lo largo de todo el trabajo. Además, se explicarán las decisiones tomadas y los criterios utilizados para garantizar la fiabilidad y validez de los resultados obtenidos. Esta sección es importante para asegurar la transparencia y replicabilidad del estudio.

6.1. Etapa de visualización

La etapa de visualización buscaba dos principales objetivos. Primero, se buscó desarrollar una manera de conseguir la representación de los canales RGB dentro de los 300 canales disponibles. Y segundo, se buscó desarrollar una manera de conseguir información hiperespectral completa referente a un único píxel o a un conjunto de píxeles dentro de la imagen, tanto la información del espectro infrarrojo, como del espectro visible como del espectro ultravioleta.

Para lograr estos objetivos se optó por el desarrollo de un pequeño software de escritorio con la finalidad de tener acceso a todos estos requerimientos de una manera cómoda e intuitiva, de manera tal que fuese un apoyo para las demás etapas que componen la investigación.

El software fue desarrollado utilizando el lenguaje de programación Python. Principalmente fueron requeridas de las librerías spectral y numpy para trabajar con los datos hiperespectrales, matplotlib para la representación de los gráficos con información hiperespectral y pyqt para la interfaz gráfica.

Código disponible en: https://github.com/Felipe1401/hyperspectral-viewer

6.1.1. Representación RGB

El software permite visualizar una representación en RGB de la imagen hiperespectral, donde se seleccionan tres rangos de longitudes de onda asociadas al color representado. Para el color rojo se seleccionó el rango de longitud de onda desde 630 a 720 nanómetros, para el color

verde se seleccionó el rango de longitud de onda desde 495 a 530 nanómetros y para el color azul se seleccionó el rango de longitud de onda desde 425 a 480 nanómetros.

En el caso del rojo, el rango de longitudes de onda se compone de 42 valores, para el color verde, el rango seleccionado se compone de 16 valores y finalmente, para el color verde hay un total de 26 valores. Con estos valores se calcula un promedio simple para obtener la representación más precisa de cada uno de los colores.

En resumen, dentro de los 300 canales son seleccionados tres rangos de canales asociados a rojo, verde y azul para poder obtener un promedio simple de estos valores y así, conseguir los canales rojo, verde y azul para ser utilizados como una representación RGB de la imagen hiperespectral con la cual se está trabajando.

En la figura 5 se ejemplifica esta característica del software. En ella, se realizó una selección manual de tanto de un archivo .bil como de un archivo .bil.hdr y posteriormente se seleccionó la opción de "representación RGB" en el tercer botón de las opciones del panel central.

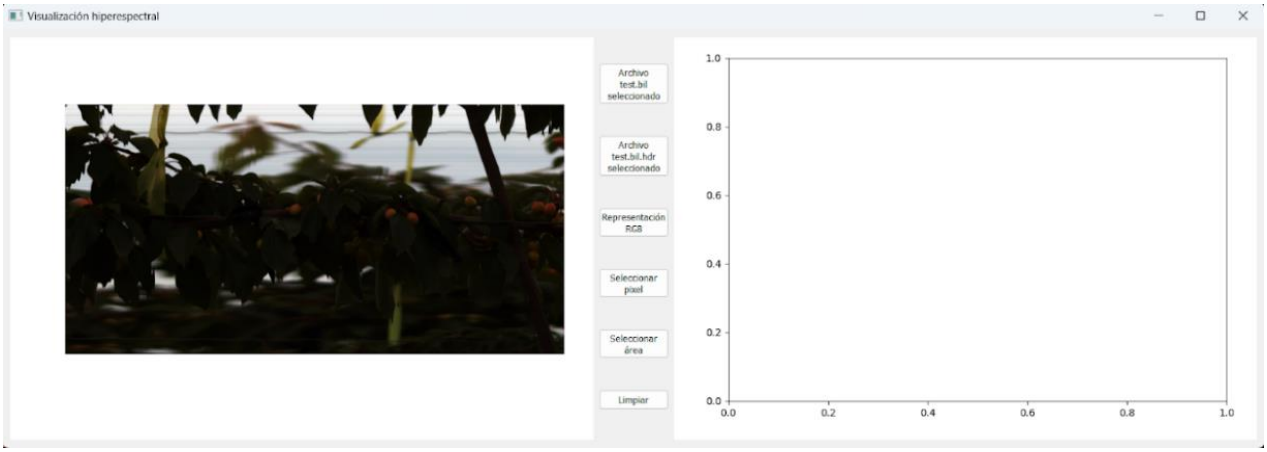

Figura 5. Representación RGB en software de visualización

6.1.2. Información hiperespectral por píxel

El software desarrollado, además, presenta una funcionalidad que facilita la visualización e interpretación de los datos que conforman el espectro hiperespectral asociados al píxel solicitado mediante una representación gráfica donde cada longitud de onda se visualiza con su color respectivo y el espectro infrarrojo se visualiza de color gris.

La figura 6 ejemplifica esta característica del software. En ella, se realizó una selección manual de un píxel dentro de la representación RGB de la izquierda, para así obtener cada una de sus componentes hiperespectrales en la visualización del gráfico derecho.

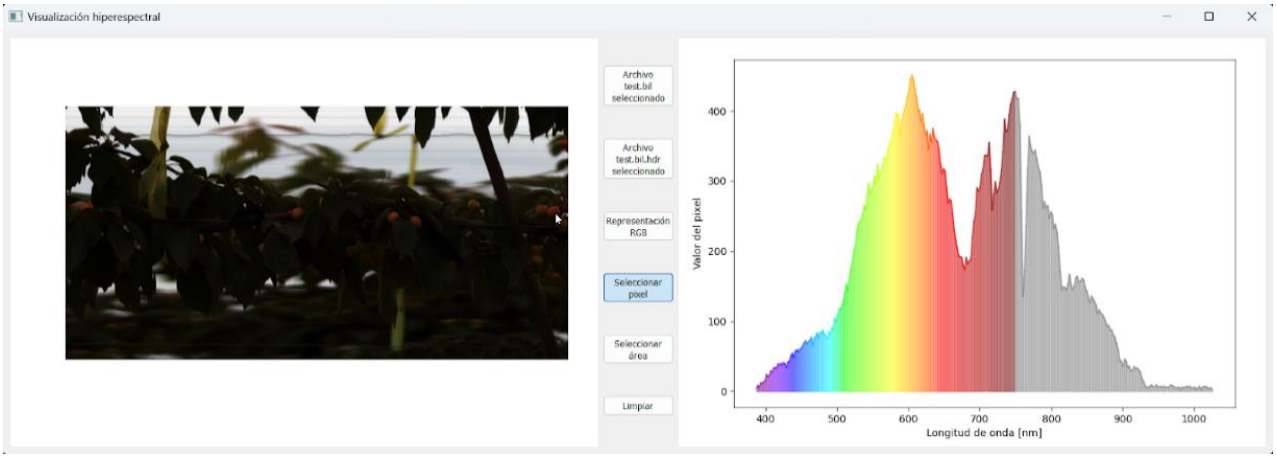

Figura 6. Información del píxel seleccionado.

6.1.3. Información hiperespectral por área

El software también incorpora una función que permite a los usuarios seleccionar libremente un área específica dentro de la representación RGB de la imagen hiperespectral. Esta selección habilita la obtención de un gráfico detallado, análogo al generado para un único píxel, pero aplicado a la región elegida. En esta área, el software resalta los valores mínimos, promedio y máximo, facilitando un análisis exhaustivo del área de interés.

La figura 7 ilustra de manera ejemplar esta característica del software. En ella, se muestra cómo se seleccionó un área específica, delimitada por un círculo blanco en la representación RGB de la izquierda, y se representó su información hiperespectral en el gráfico a la derecha. En este gráfico, se distinguen claramente tres líneas remarcadas: la inferior, representando todos los datos mínimos; la central, indicando los valores promedio; y la superior, mostrando los valores máximos de la región seleccionada.

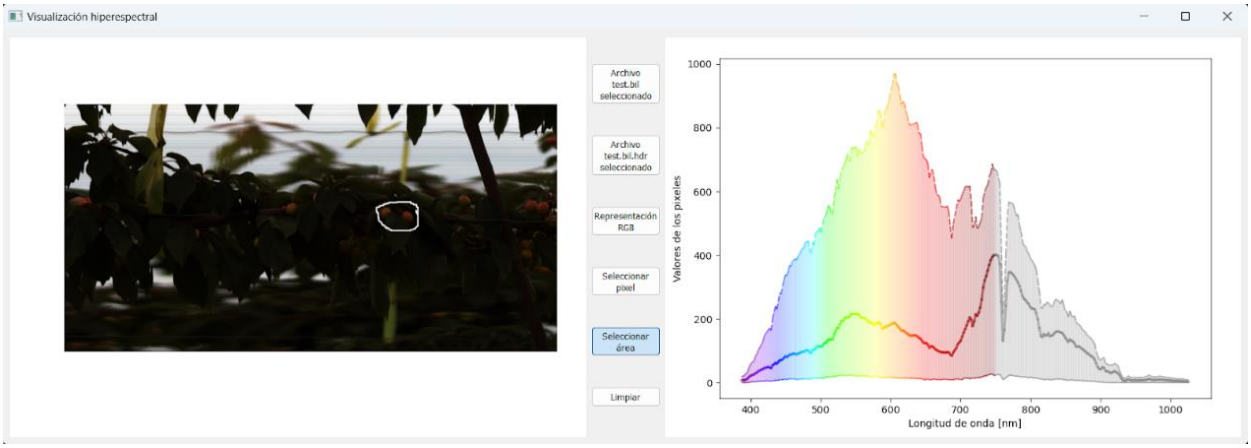

Figura 7. Información del área seleccionada.

6.1.4. Limpiar información

Finalmente, una funcionalidad clave del software desarrollado fue la inclusión del botón nombrado como "limpiar" (se puede observar en el último botón de la figura 7.), el cual permite deseleccionar el píxel o área de píxeles retornando así la representación RGB de la imagen hiperespectral a su estado original y dejando el gráfico de la derecha en blanco (como en la figura 5).

6.2. Etapa de segmentación

Durante la etapa de segmentación se utilizaron diversos enfoques de manera exploratoria con el fin de conseguir los mejores resultados, es decir, los mejores conjuntos o regiones de píxeles con características similares. En el caso particular con los datos utilizados, se definirá como regiones de interés en la imagen a las cerezas.

El marco general de trabajo fue iniciar con un preprocesamiento de los datos para reducir la dimensionalidad de la matriz de píxeles a trabajar y posteriormente usar estos datos con dos enfoques de segmentación diferentes, con algoritmos clásicos y con métodos de aprendizaje no supervisado.

Toda la etapa de segmentación fue trabajada utilizando Python complementado de librerías como Pillow, Opencv, Matplotlib para visualización de las imágenes, Numpy y Scipy para el manejo de las estructuras de datos realizadas, sikit-learn para el uso de los modelos de

aprendizaje no supervisados y la librería spectral para el uso de archivos de imágenes hiperespectrales.

6.2.1. Preprocesamiento de los datos

El objetivo principal de la etapa preprocesamiento de los datos fue reducir la dimensionalidad de las imágenes hiperespectrales, debido a que sus tamaños son aproximadamente de 1800x900 píxeles, y lo que sumado a que cada píxel cuenta con 300 canales de información hiperespectral, nos deja con un total de 486.000.000 datos aproximadamente para trabajar por imagen.

Se seleccionó la técnica de análisis de componentes principales (PCA) para la reducción de dimensionalidad debido a su simpleza y rapidez, en pro de favorecer los tiempos de ejecución y la simplicidad del proceso de preprocesamiento de los datos. Se entrenó un PCA para cada una de las imágenes, en donde se utilizaron todos los píxeles de la imagen.

Para garantizar el óptimo funcionamiento del análisis de componentes principales se utilizó una técnica llamada escalador estándar (standard scaler) que normaliza los datos con el fin de que tengan una media de 0 y una desviación estándar de 1, asegurando así, que todos los canales de información contribuyan de manera equivalente en el análisis posterior.

El uso de el escalador estándar es crucial, debido a la sensibilidad del análisis de componentes principales a los datos que tienen una mayor varianza, lo que puede influir desproporcionadamente en los resultados.

Al estandarizar los datos, se asegura que todas las variables contribuyan de manera equivalente en el análisis.

Para seleccionar la cantidad de componentes principales a utilizar, se realizó un análisis de la covarianza acumulada por componentes, donde se buscó al menos un 95% para garantizar una representación lo suficientemente precisa y que obtendría resultados lo más idénticos posibles a la utilización de la cantidad de datos originales.

Finalmente, se seleccionaron 30 componentes principales como base de datos transformada para trabajar, reduciendo así la dimensionalidad de los datos de 300 a 30, por lo

cual se trabajó con una cantidad total de 48.600.000 datos en lugar de los 486.000.000 datos originales.

6.2.2. Segmentación con algoritmos clásicos

6.2.2.1. Segmentación utilizando distancia euclidiana.

El primer método empleado para conseguir una segmentación parcial de la imagen fue utilizando la distancia euclidiana como criterio de similitud entre cada píxel. Para esto, se seleccionó un píxel representativo del área de interés a segmentar, para así calcular la distancia que existe entre un par de puntos dentro del conjunto y utilizarlo como un umbral de referencia para determinar si un punto pertenece o no al segmento buscado. Al umbral se le consideró un poco de holgura para poder captar de mejor manera la totalidad de los píxeles buscados.

La distancia euclidiana, que básicamente mide la distancia "en línea recta" entre dos puntos, en este caso puntos dados por las 30 componentes principales seleccionadas, se calculó utilizando la formula estándar, es decir, calculando la raíz cuadrada de la suma de los cuadrados de las diferencias entre las coordenadas de ambos puntos.

Este enfoque fue importante para lograr un primer acercamiento a la segmentación en imágenes, lo que fue crucial para permitir trazar la ruta de métodos a evaluar en el resto de la investigación.

6.2.2.2. Segmentación utilizando watershed.

El segundo método utilizado para la segmentación de imágenes fue el algoritmo watershed. En esta implementación se utilizó la distancia de Mahalanobis como el criterio de similitud para evaluar entre los píxeles. Esta elección permitió una evaluación más precisa y diferenciada entre los píxeles, facilitando una segmentación más efectiva.

La implementación de este método requirió una selección manual cuidadosa de píxeles semilla, que sirvieron como puntos de referencia para los criterios de comparación durante la segmentación. Estas semillas se escogieron específicamente de entre los píxeles de algunas de las cerezas más llamativas y visibles en la imagen. La elección de semillas es crucial para el desempeño del algoritmo.

Es importante señalar que la replicabilidad de los resultados obtenidos depende en gran medida de las semillas elegidas, por lo cual, existe un gran grado de certeza de que en distintas ejecuciones (variando las semillas) se obtengan diferentes áreas segmentadas.

6.2.3. Segmentación con métodos de aprendizaje no supervisados.

6.2.3.1. Segmentación utilizando HDBSCAN.

Para utilizar el enfoque de HDBSCAN, se entrenó un modelo del algoritmo con cada una de las imágenes disponibles en el conjunto de prueba de manera individual, de esta manera, se obtuvieron cuatro modelos entrenados específicamente para cada una de las imágenes, con el objetivo de obtener los modelos más precisos para el caso particular en el que se está trabajando.

Para este enfoque se utilizó una librería que implementa el algoritmo HDBSCAN (McInnes, L., Healy, J., & Astels S, 2016), la cual está basada en el artículo científico "Accelerated Hierarchical Density Based Clustering" (McInnes & Healy, 2017), que a su vez está basado en las clases implementadas dentro de la librería Sikit-learn, la cuál es ampliamente conocida para el análisis de datos con Python.

A raíz de esto se obtienen diversas cantidades de clusters para cada una de las imágenes del conjunto de prueba, las cuáles son presentadas en un gráfico.

Inicialmente se optó por utilizar el algoritmo DBSCAN, sin embargo, dado los resultados obtenidos, se optó por la utilización de HDBSCAN en pro de un algoritmo que fuera más versátil respecto a la variabilidad de los datos en términos de densidad y más adaptable a una alta dimensionalidad de datos, ya que, a pesar de reducir de 300 a 30 canales, se debe tener en cuenta la alta dimensionalidad de los datos.

6.2.3.2. Segmentación utilizando k-means.

Se entrenó un modelo de k-means para cada una de las imágenes del conjunto de prueba, donde se utilizó el método k-means++ para la inicialización de los centroides debido a su mejor desempeño, pero se varió la cantidad de centroides de inicialización entre 1 a 10 y, en consecuencia, la cantidad de grupos generados.

Con esto, se realizó un análisis de la cantidad de clustering utilizando el método del codo, donde se busca analizar la inercia en la cantidad de clusters evaluado y seleccionar donde se encuentre el punto de inflexión del gráfico (Syakur, Khotimah, Rochman, & Satoto, 2018). Siguiendo este enfoque, se determinó que la cantidad más representativa para el conjunto de imágenes analizadas es de cuatro clusters.

Posteriormente se realizaron las visualizaciones correspondientes para los cuatro segmentos encontrados dentro de la imagen. Adicionalmente se realizó una comparación utilizando k-means con cuatro clusters aplicado a una imagen de laboratorio en su versión RGB e hiperespectral con el fin de comprender la efectividad de las imágenes hiperespectrales.

6.3. Etapa de cuantificación.

En la etapa de cuantificación se realizó un análisis comparativo entre las regiones encontradas en los enfoques de aprendizaje no supervisado. Se analizó la cantidad de regiones encontradas en cada una de las imágenes del conjunto de imágenes de prueba. Además, se analizaron sus tamaños en base al espacio utilizado dentro de la imagen completa para obtener una comparación entre los funcionamientos de ambos algoritmos.

Es necesario aclarar que hablar de “regiones” es análogo decir “cluster” pero en un contexto del espacio que utiliza este dentro de la imagen original.

7. Análisis de resultados

En esta sección se presentarán los resultados obtenidos a través del procedimiento previamente descrito en la sección de Marco Metodológico. Se presentará el conjunto de datos utilizado y, además, se realizará un análisis claro y consistente de cada uno de los resultados en cada una de las etapas de la investigación.

7.1. Datos utilizados

El set de datos utilizado se compone de diferentes imágenes hiperespectrales capturadas usando la cámara "Pika L" de Resonon. El largo de cada imagen se mantiene en 900 píxeles, ya que es el máximo permitido por la cámara, sin embargo, el ancho puede variar dependiendo de lo que se requiera.

Cada píxel dentro de la imagen se compone por 300 canales de información sobre la longitud de onda que se logró capturar. La información de longitud de onda se mantiene en un rango de los 400 a los 1000 nanómetros, lo que corresponde a información dentro del espectro visible (400 nm - 700 nm) y del espectro infrarrojo (700 nm - 1000 nm).

7.1.2. Imágenes de cerezas

El conjunto de imágenes hiperespectrales utilizado para las pruebas de segmentación está conformado por cuatro imágenes tomadas en terreno utilizando la cámara hiperespectral Pika L de Resonon. Una de las imágenes fue tomada en un cultivo de cerezas en Requínoa, dos imágenes fueron tomadas en un cultivo de cerezas Rengo y una imagen fue tomada en un cultivo de cerezas en Graneros.

Es importante aclarar que las representaciones en RGB de las imágenes hiperespectrales no son representativas de las condiciones en las que las imágenes fueron capturadas, ya que se les aplicó un tratamiento de brillo y contraste para facilitar la visualización de los elementos que se buscaban capturar cuando la fotografía fue realizada.

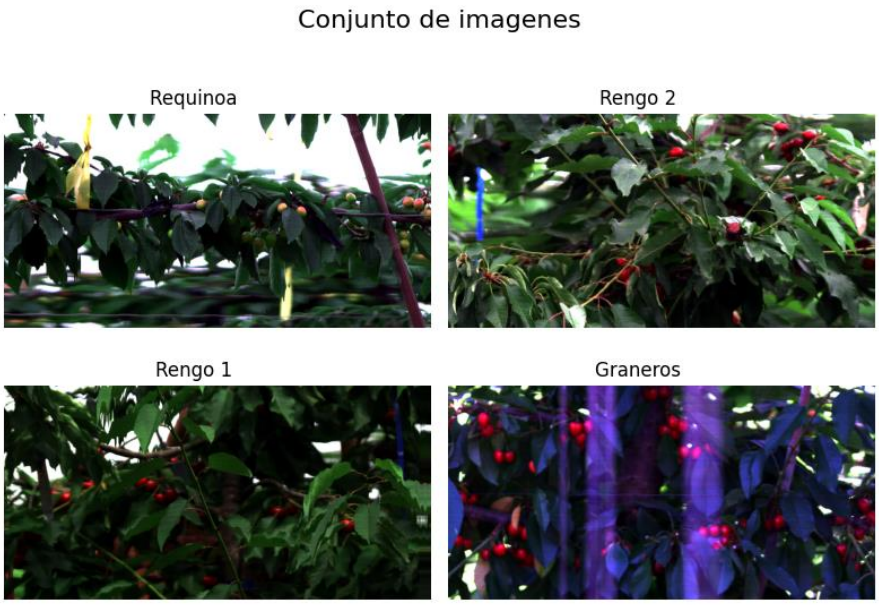

Figura 8. Representación RGB del conjunto de imágenes utilizadas.

Adicionalmente se utilizó una imagen hiperespectral tomada en un laboratorio, es decir, en un ambiente controlado donde la luminosidad y posiciones de las cerezas fueron cuidadosamente seleccionadas. Con estas características es más fácil ejemplificar el funcionamiento de algunos métodos de segmentación. Esta imagen de igual manera fue capturada con la cámara Pika L de Resonon.

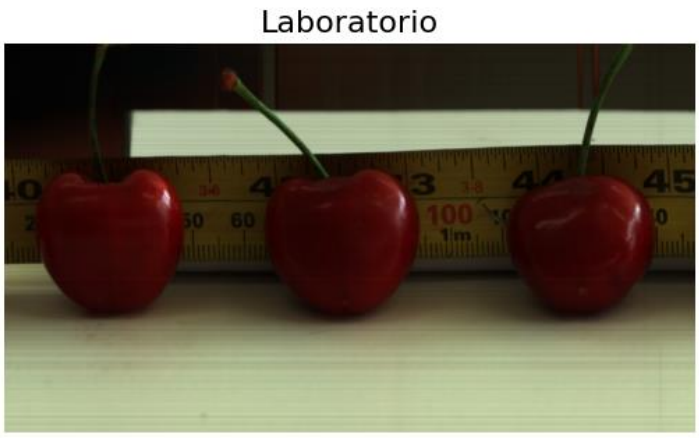

Figura 9. Representación RGB de imágenes utilizadas en entorno controlado.

7.2. Resultados obtenidos

7.2.1. Análisis PCA

El análisis sobre la cantidad de componentes óptimos para utilizar, aplicado a 3 imágenes diferentes del conjunto de datos disponibles, muestra que a partir de 13 componentes principales se logra alcanzar una representatividad promedio del 95% (ver Fig. 11). Sin embargo, con 30 componentes se obtiene una tasa superior al 98% de representatividad para ambas imágenes en contexto de cultivos, por lo cual, esta fue la cantidad de componentes principales elegida para trabajar durante el resto de la investigación.

Es importante notar que la imagen que alcanza una menor varianza acumulada es una imagen de cerezas maduras tomada en condiciones de laboratorio, por lo cual, no es representativa al contexto de la mayoría de las imágenes, ya que estas fueron tomadas directamente desde los cultivos de cerezas.

Por esta razón, es posible extrapolar el comportamiento obtenido en este análisis para justificar el uso de 30 componentes principales en el proceso de segmentación de cualquier imagen en el contexto de terreno, ya que, una representación del 95% es considerada más que aceptable para conseguir resultados tremendamente similares al uso de los 300 canales de información.

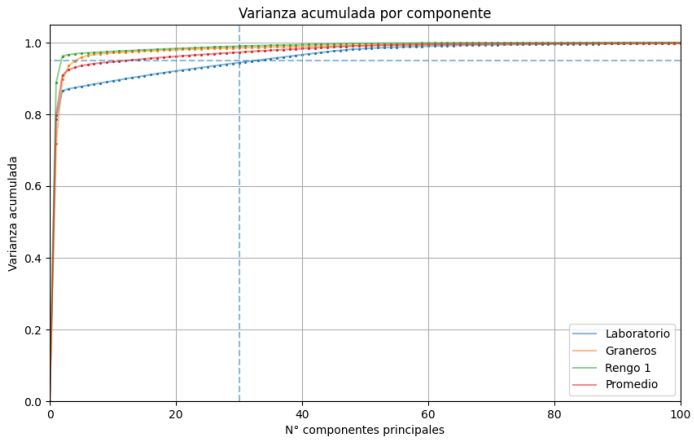

Figura 10. Varianza acumulada por componentes.

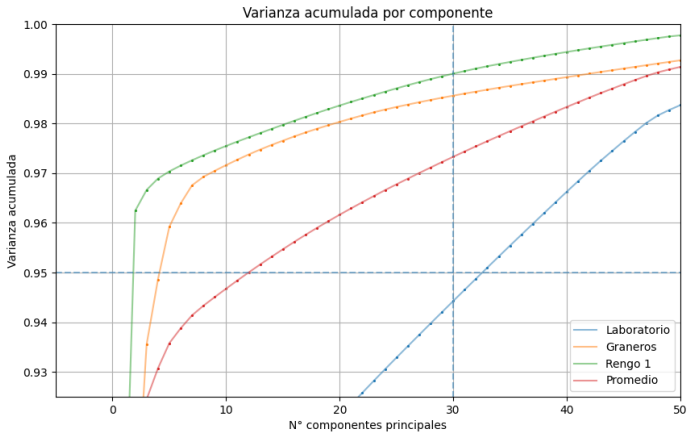

Figura 11. Zoom figura 10.

Reducir de 300 canales a 30 componentes ofrece dos beneficios cruciales para el desarrollo de la investigación. Primero, al reducir la cantidad de datos para trabajar se reduce significativamente el tamaño en memoria requerido para almacenarlos. Dado que las variables son almacenadas en la memoria RAM del computador durante el proceso de ejecución, se logra evitar el que el sistema operativo deba abortar los procesos al llenar la memoria RAM. Y segundo, los tiempos de ejecución se reducen considerablemente, ya que, se está logrando reducir la cantidad de datos 10 veces.

En resumen, tanto el almacenamiento como los tiempos de ejecución se ven beneficiados por la reducción del tamaño de los datos, lo que permite trabajarlos con mayor facilidad y consiguiendo resultados casi idénticos a trabajar con la cantidad de datos original.

Para un análisis más riguroso de la eficacia del análisis de componentes principales revisar el anexo 1.

7.2.2. Análisis segmentación con distancia euclidiana.

La imagen seleccionada para las pruebas de la segementación con enfoque de distancia euclidiana fue la correspondiente al cultivo de cerezas de Requínoa, ya que, es la imagen que presenta mayor variación de color en las cerezas, por lo que fue un caso de estudio mucho más

interesante. Además, el píxel objetivo seleccionado se encuentra dentro de uno de las cerezas, por lo que se esperó lograr una segmentación de las cerezas en la imagen.

Como se pudo observar en la imagen (ver Fig. 12), ambos resultados fueron bastante similiares, sin embargo, la imagen hiperespectral presentaron pequeñas mejoras que puden ser determinantes para en algunos casos. Por ejemplo, en la esquina superior derecha se puede observar una hoja de color similar a las cerezas, la cuál en la segmentación RGB es mucho más confundida como segmento de cereza. Podemos observar como gracias a la información del espectro infrarrojo, para el caso de la imagen hiperespectral es posible descartarla con mayor precisión. Además, se pueden observar algunas cerezas que fueron detectadas en la imagen hiperespectral, más sin embargo, no en la imagen RGB.

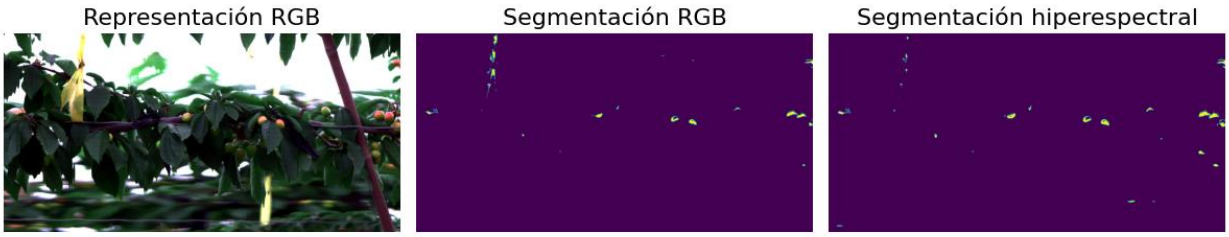

Figura 12. Distancia euclidiana para imagen de Requínoa.

7.2.3. Análisis segmentación con watershed.

Se aplicó el algoritmo watershed únicamente sobre la imagen de cultivo de cerezas de Requínoa por la misma razón que se aplicó en la segmentación utilizando distancia euclidiana. Se eligieron píxeles pertenecientes al segmento de la cereza que se logra ver en la imagen RGB

Se obteniendo los siguientes resultados (ver Fig. 13), en donde podemos observar que la imagen con información RGB logra segmentar únicamente la cereza a la que le pertenecen los puntos, sin embargo. Por otro lado, la imagen con información hiperespectral logra segmentar de mejor manera algunas de las cerezas que se encuentran presente dentro de la imagen, sin embargo, tal como en el caso enfoque anterior, hay problemas con las hojas amarillas, las cuáles

posiblemente tienen una firma hiperespectral (color e infrarrojo) similar a los píxeles utilizados como semilla.

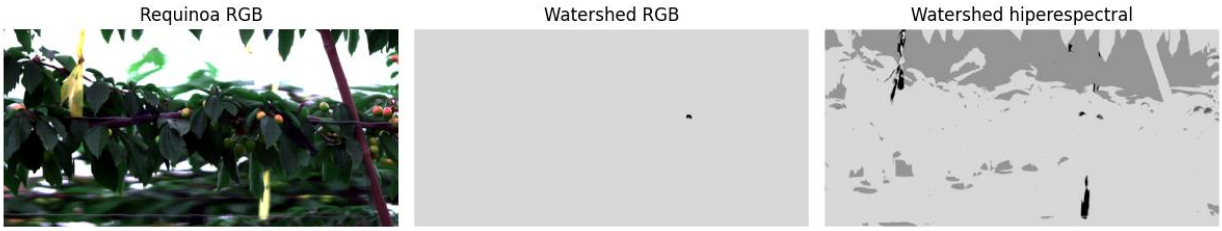

Figura 13. Watershed para imagen de Requínoa.

7.2.4. Análisis segmentación con HDBSCAN.

Los resultados obtenidos en cada una de las imágenes del conjunto de datos muestran cómo claramente el enfoque de HDBSCAN no es para nada efectivo, al menos para el problema trabajado, dado que está segmentando de manera poco eficaz. En todos los casos se están considerando casi la totalidad de los píxeles dentro de una misma región, lo que no refleja ningún área de interés en particular.

Además de esto, los tiempos de ejecución fueron demasiado elevados, rondando en torno a las 12 horas por imagen para completar el procesamiento.

El algoritmo fue definido para que logre encontrar al menos 5 clusters, por lo cual, la configuración de los parámetros dados puede ser uno de los causantes de los malos resultados, sin embargo, existen otros factores involucrados que pudieron haber afectado. La dimensionalidad de los datos analizados es de 30 (por el análisis de componentes principales). Esta alta dimensionalidad puede estar entorpeciendo el desempeño del algoritmo.

Los resultados obtenidos con la utilización de este algoritmo en el contexto de cultivo de cerezas nos permiten descartarlo como un algoritmo efectivo para lograr la correcta segmentación en imágenes hiperespectrales.

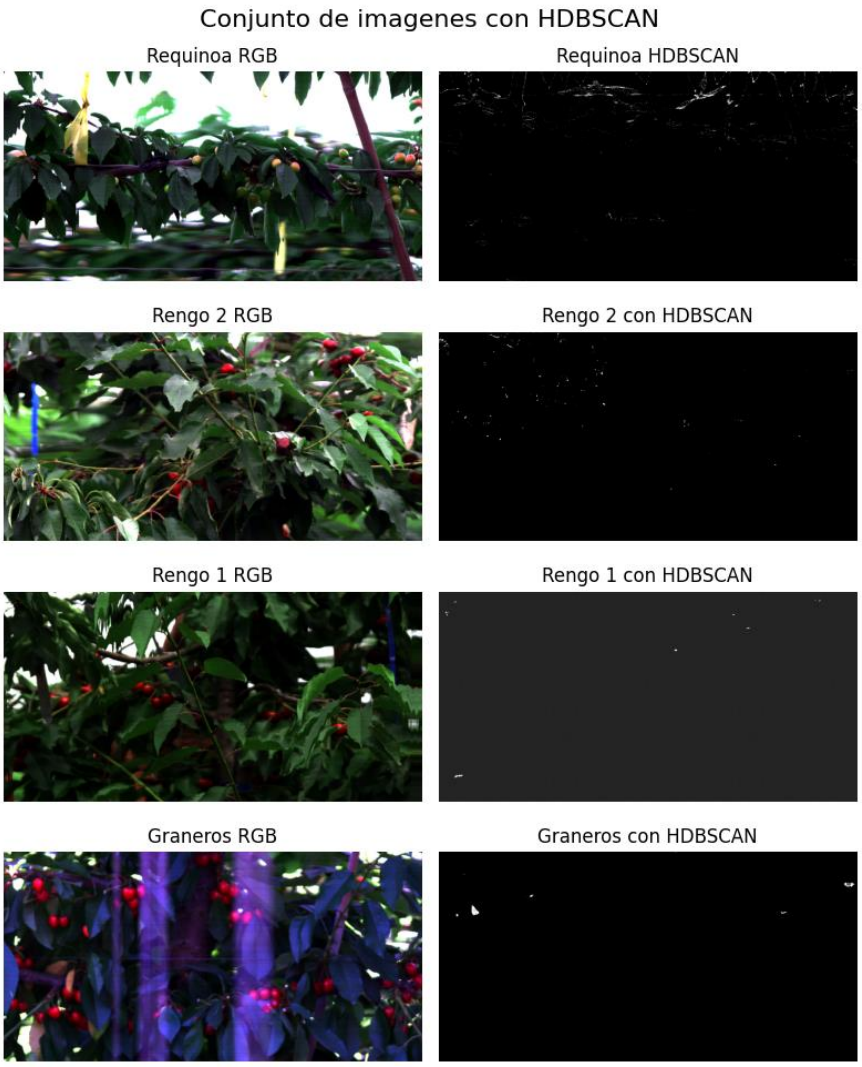

Figura 14. HDBSCAN para el conjunto de imágenes.

7.2.5. Análisis segmentación con k-means.

El análisis del codo determinó que el punto de inflexión en el gráfico (ver fig. 15) se encuentra en el k igual a cuatro, por esta razón, la cantidad de clusters elegida para el análisis de los datos será de cuatro.

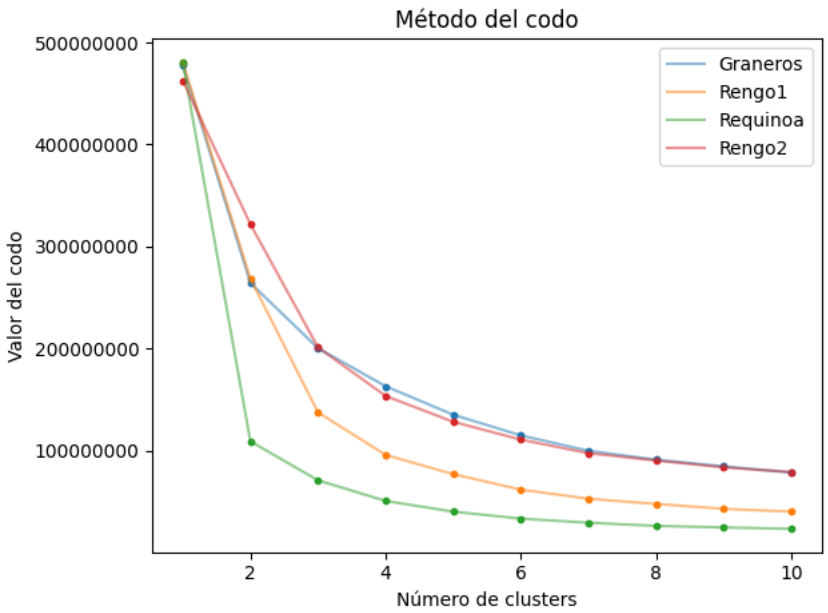

Figura 15. Método del codo aplicado al conjunto de imágenes.

La segmentación está representada en la visualización con un color para cada uno de los clusters o zonas segmentadas, distinguiéndose así unas de otras (ver Fig. 16).

La visualización de los clusters o áreas segmentadas tanto para la imagen del cultivo de cerezas de Requínoa como las del cultivo de cerezas de Rengo, muestran claramente que se está destinando dos clusters para la vegetación presente y dos clusters para la información captada por el fondo. En el caso de los clusters de la vegetación, se está dividendo entre dos segmentos, los cuáles podrían reflejar un estado de salud más similar (basado en su información del espectro infrarrojo) entre los elementos pertenecientes a los clusters respectivos.

Por el lado de la imagen cultivo de cerezas de Graneros se observa una clara dificultad a la hora de clasificar los píxeles dentro de la imagen, esto puede deberse a la baja condición lumínica cuando fue tomada la fotografía o a la presencia de sombras de mayor intensidad, ya que como se mencionó anteriormente, las representaciones RGB utilizadas fueron tratadas con variaciones de contraste y brillo para poder visualizar con mayor facilidad lo que se intentó capturar en el cultivo.

A pesar de esta última situación, el enfoque utilizado con k-means (k = 4) parece ser el enfoque con mejores resultados, dado que logra segmentar, en su mayoría, la vegetación y el fondo presente en las imágenes.

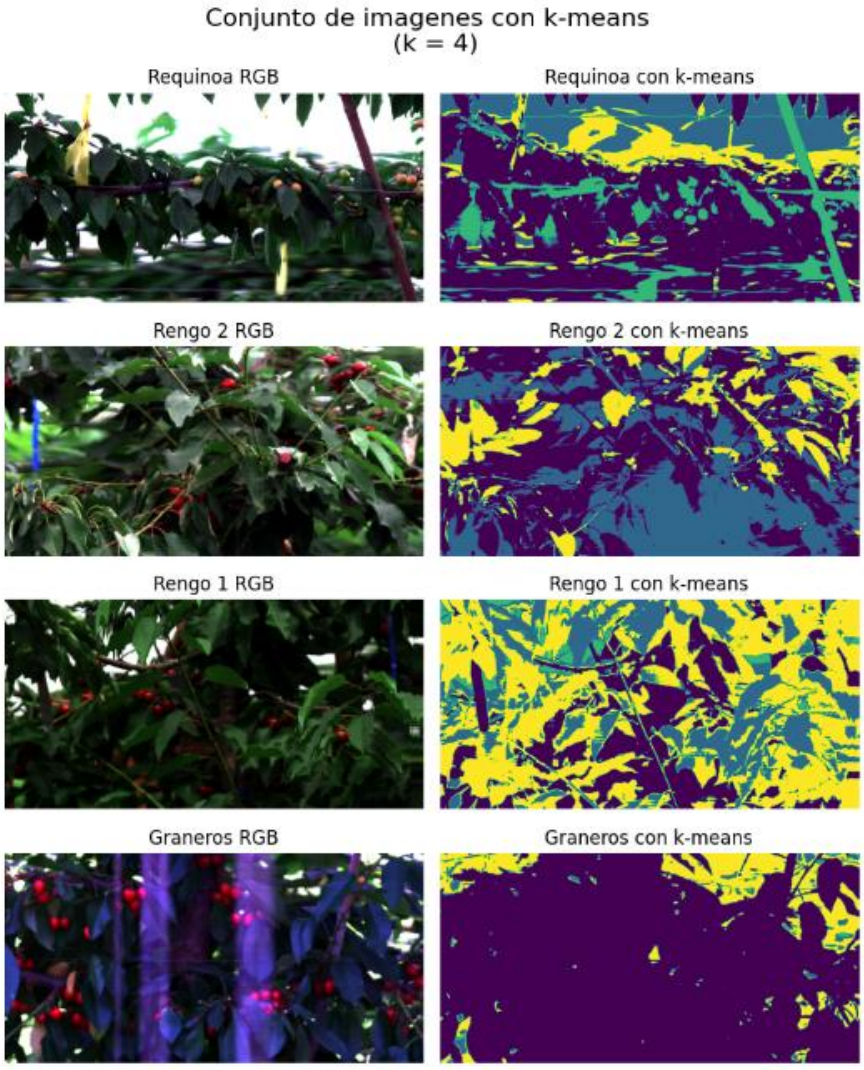

Figura 16. k-means (k = 4) para el conjunto de imágenes.

Para analizar la efectividad de esta técnica, no sólo en comparación a las demás técnicas, si no que, además, en comparación con información entregada únicamente por una imagen RGB, se realizó la comparativa en donde se utilizó la imagen en condiciones de laboratorio.

Como se puede observar en el caso de la imagen con representación RGB (ver Fig 17.) se obtiene una segmentación muy ineficiente e ineficaz de la cereza, la cuál es la única zona de

interés para segmentar dentro de esta imagen, donde se confunden elementos como los números de la regla de medición con las cerezas en el cluster número 3.

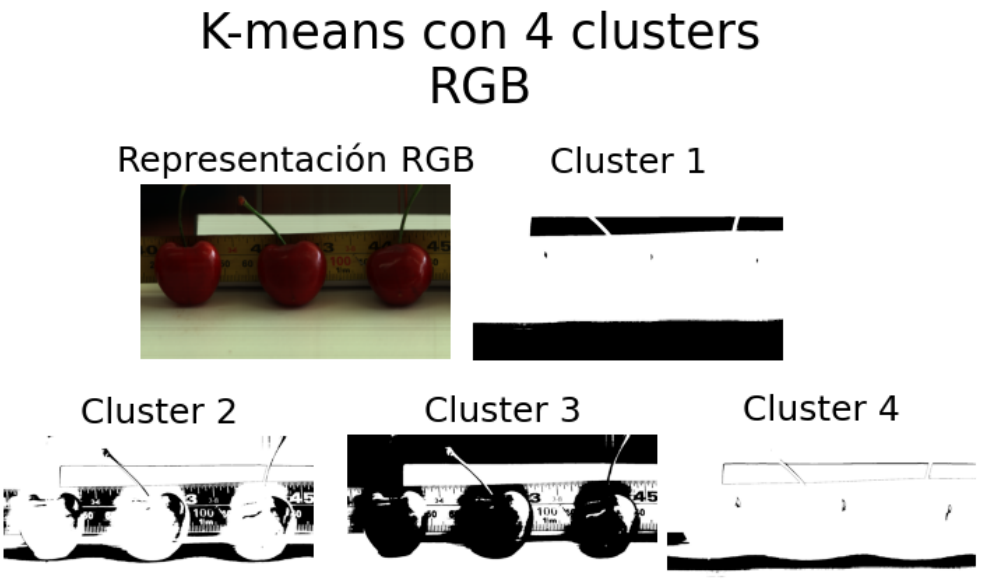

Figura 17. k-means (k = 4) para imagen de laboratorio en RGB.

Por otro lado, en el caso de la imagen que cuenta con información hiperespectral (ver Fig. 18), se obtuvieron segmentaciones mucho más precisas, esto debido a que la suma de la información del espectro visible y la información del espectro infrarrojo es inconfundible para cuando se trata de encontrar elementos que “tienen vida”, como en este caso, las cerezas.

Como se puede observar, en este caso, el cluster 3 no tiene ningún problema para conseguir una segmentación adecuada de las cerezas dentro de la imagen, debido a, como se mencionó anteriormente, la diferencia del espectro infrarrojo que tienen los elementos.

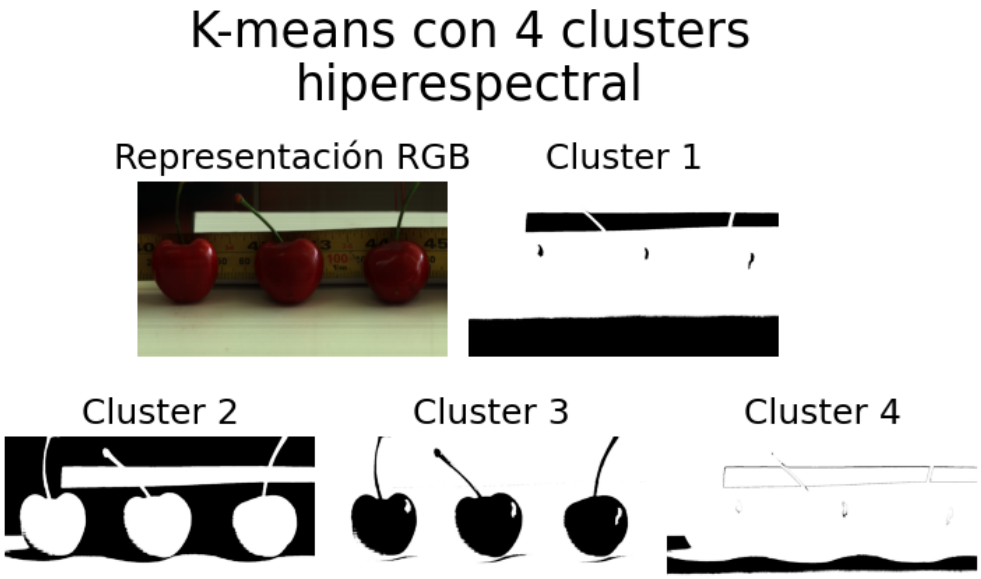

Figura 18. k-means (k = 4) para imagen de laboratorio en hiperespectral.

Con este análisis queda en evidencia la eficacia de utilizar imágenes hiperespectrales por sobre imágenes con información únicamente en tres canales (RGB). La gran cantidad de información dentro del hiperespectro es clave para lograr una segmentación mucho más precisa en casi cualquier contexto posible, como en este caso por ejemplo con la agronomía.

Finalmente, es de destacar que los tiempos de ejecución fueron considerablemente bajos respecto a los demás enfoques realizados. Por ejemplo, el análisis del método del codo requirió del entrenamiento de un total de cuarenta modelos de k-means con una cantidad de clusters que iba desde uno a diez y tomó un tiempo de ejecución de alrededor de 45 minutos.

7.3. Análisis de cuantificación.

Iniciando con el algoritmo de HDBSCAN, este obtuvo resultados poco consistentes con la segmentación esperada, en donde se obtuvieron los siguientes resultados. En la siguiente tabla se verán reflejados la cantidad de clusters obtenidos y el porcentaje que ocupa el cluster que representa la región más grande de la segmentación.

Imagen segmentada	Número de clusters	Tamaño de la mayor región
Graneros	9	99.97%
Rengo 1	36	99.90%
Rengo 2	5	99.91%
Requínoa	5	99.53%

Tabla 1. Cantidad de cluster y región más significativa de HDBSCAN.

Todos los resultados obtuvieron un cluster con representación superior al 99.5% de los datos, por lo cual, los demás clusters presentas regiones insignificantes en comparación. Esto es concordante con los datos visualizados en la figura 14, en donde se pude ver claramente la predominancia de un color (que representa a un cluster) por sobre el resto.

Por el lado de k-means se obtuvieron resultados mucho más representativos de zonas de interés dentro de la imagen, lo que quedó más que demostrado dentro del análisis realizado con los resultados obtenidos en la figura 18.

En la siguiente tabla se analizarán los porcentajes que cada región representa dentro de la imagen.

Imagen segmentada	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Graneros	87.65%	1.18%	11.04%	0.13%
Rengo 1	49.84%	0.07%	32%	17%
Rengo 2	24.44%	1.95%	50.30%	23.31%
Requínoa	11.97%	57.2%	16.82%	14%

Tabla 2. Porcentaje de región o cluster en k-means.

En este caso, la cantidad de clusters es fija, dado que, fue seleccionada en base al análisis realizado con el método del codo.

Como vemos, cada uno de los porcentajes es concordante con la visualización entregada en la figura 16, además, se pudo observar que, en las imágenes de Graneros, Rengo 1 y Rengo 2

se obtuvieron regiones que representan a menos del 2% de la imagen. Esto puede indicar que la elección de la cantidad de clusters fue la correcta, ya que, se está capturando un grupo de datos que no es tan representativo dentro de la imagen, es decir, el resto de los clusters sí están muy bien agrupados.

Con estos datos, queda más que claro que el enfoque con mejor desempeño es el algoritmo de k-means, el cual logra una mejor distribución respecto a los porcentajes que utiliza cada una de las regiones encontradas.

8. Conclusión y trabajo futuro

8.1. Conclusiones generales

En conclusión, la investigación y trabajo desarrollado dejó en evidencia las limitaciones que se deben sortear para conseguir una segmentación sin la ayuda de un conjunto de datos previamente etiquetado. La investigación mostró que k-means, con un correcto análisis de clusters a utilizar, es una opción más que aceptable para la segmentación de imágenes hiperespectrales, tanto en términos de eficacia en los resultados, como en términos de tiempos de ejecución a la hora de entrenar y utilizar un modelo.

Además, se logró demostrar que las imágenes hiperespectrales en el contexto de segmentación de imágenes, resultan ser mucho más valiosas gracias a su gran cantidad de datos en el hiperespectro, consiguiendo un mucho mejor desempeño que las imágenes tradicionales en tres canales (RGB).

El principal desafío enfrentado fue sin lugar a dudas la falta de una base de datos más completa y etiquetada, ya que sumado al tiempo acotado para el desarrollo de la investigación, no permitió considerar un enfoque de aprendizaje supervisado, sin embargo, los resultados obtenidos por k-means, en particular en un ambiente controlado, demostraron que existe un enfoque factible y prometedor que debe seguir siendo explorado para obtener resultados similares en objetos más pequeños y en la intemperie.

Los resultados entregados por k-means respaldan la hipótesis planteada en la sección 2, la que proponía lograr una segmentación más precisa utilizando una combinación de técnicas de visión computacional y aprendizaje de máquina con información hiperespectral en comparación a la segmentación utilizando únicamente información RGB. En ambientes controlados, la información hiperespectral demostró ser un instrumento realmente valioso para conseguir la identificación de áreas de interés. Este hallazgo resulta potencialmente significativo para generar nuevas perspectivas y oportunidades para futuras investigaciones en el ámbito de la visión por computadora y el procesamiento de imágenes.

Por último, la investigación llevó a desarrollar un software de escritorio de código abierto desarrollado en Python que permite la visualización de imágenes hiperespectrales. El software está disponible en GitHub de manera abierta para cualquier investigación futura que lo requiera.

8.2. Trabajo futuro

Respecto al trabajo futuro, los resultados obtenidos permiten dar una base clara sobre la segmentación de imágenes hiperespectrales con enfoque de aprendizaje de máquina no supervisado, por lo cual, sería posible comenzar a explorar nuevos métodos con diferentes enfoques, por ejemplo, un método semi supervisado que permita reentrenar modelos en base a la retroalimentación manual entregada por el usuario.

Además, se podrían explorar más y mejores técnicas de reducción de dimensionalidad, para lograr un mayor nivel de representatividad. Actualmente existen modelos de autoencoders especialmente diseñados para trabajar con imágenes hiperespectrales que tienen una muy alta dimensionalidad, por lo cual adaptar uno al tópico de cultivos de cereza no debería resultar particularmente desafiante.

Durante el desarrollo de esta investigación se comenzó a experimentar con el uso de autoencoders para aprovechar el encoder y así lograr la reducción de dimensionalidad, el modelo logró ser entrenado con 400.000 píxeles aleatorios pertenecientes al conjunto de imágenes con el que se trabajó. Sin embargo, la idea fue descartada por limitaciones temporales.

9. Referencias

Lucasbosch. (2021). Spectral sampling RGB multispectral hyperspectral imaging [Imagen]. Wikimedia Commons. https://en.m.wikipedia.org/wiki/File:Spectral_sampling_RGB_multispectral_hyperspectral_imaging.svg

Artacho, B., & Savakis, A. (2019). Waterfall Atrous Spatial Pooling Architecture for Efficient Semantic Segmentation. En Sensors (Vol. 19, Issue 24), Figura 1. MDPI AG. https://doi.org/10.3390/s19245361

Massi, M. (2019). Autoencoder schema [Imagen]. Wikimedia Commons. https://en.m.wikipedia.org/wiki/File:Autoencoder_schema.png

Helmenstine, A. M. (2020, April 1). Wavelengths and Colors of the Visible Spectrum. ThoughtCo. https://www.thoughtco.com/understand-the-visible-spectrum-608329

Fournier, Q., & Aloise, D. (2019). Empirical Comparison between Autoencoders and Traditional Dimensionality Reduction Methods. En 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE) (pp. 211–214). Sardinia, Italy: IEEE. https://doi.org/10.1109/AIKE.2019.00044

Kuester, J., Gross, W., & Middelmann, W. (2021). 1D-Convolutional autoencoder based hyperspectral data compression. En The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B1-2021, 15–21. https://doi.org/10.5194/isprs-archives-XLIII-B1-2021-15-2021

Zhouhan Lin, Yushi Chen, Xing Zhao, & Gang Wang. (2013). Spectral-spatial classification of hyperspectral image using autoencoders. En 2013 9th International Conference on Information, Communications & Signal Processing. 2013 9th International Conference on Information, Communications & Signal Processing (ICICS). IEEE. https://doi.org/10.1109/icics.2013.6782778

Lei, J., Li, X., Peng, B., Fang, L., Ling, N., & Huang, Q. (2021). Deep Spatial-Spectral Subspace Clustering for Hyperspectral Image. En IEEE Transactions on Circuits and Systems for

Video Technology (Vol. 31, Issue 7, pp. 2686–2697). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/tcsvt.2020.3027616

Hu, X., Li, T., Zhou, T., & Peng, Y. (2021). Deep Spatial–Spectral Subspace Clustering for Hyperspectral Images Based on Contrastive Learning. En Remote Sensing (Vol. 13, Issue 21, p. 4418). MDPI AG. https://doi.org/10.3390/rs13214418

Egaña, Á. F., Santibáñez-Leal, F. A., Vidal, C., Díaz, G., Liberman, S., & Ehrenfeld, A. (2020). A Robust Stochastic Approach to Mineral Hyperspectral Analysis for Geometallurgy. En Minerals (Vol. 10, Issue 12, p. 1139). MDPI AG. https://doi.org/10.3390/min10121139

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment Anything (Versión 1). arXiv. https://doi.org/10.48550/ARXIV.2304.02643

Mercier, G., & Lennon, M. (s. f.). Support vector machines for hyperspectral image classification with spectral-based kernels. En IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477). IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. IEEE. https://doi.org/10.1109/igarss.2003.1293752

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). k-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. En Information Sciences (Vol. 622, pp. 178–210). Elsevier BV. https://doi.org/10.1016/j.ins.2022.11.139

Sinaga, K. P., & Yang, M.-S. (2020). Unsupervised k-means Clustering Algorithm. En IEEE Access (Vol. 8, pp. 80716–80727). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/access.2020.2988796

Ranjan, S., Nayak, D. R., Kumar, K. S., Dash, R., & Majhi, B. (2017). Hyperspectral image classification: A k-means clustering based approach. En 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS). 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE. https://doi.org/10.1109/icaccs.2017.8014707

Ren, Z., Sun, L., & Zhai, Q. (2020). Improved k-means and spectral matching for hyperspectral mineral mapping. En International Journal of Applied Earth Observation and Geoinformation (Vol. 91, p. 102154). Elsevier BV. https://doi.org/10.1016/j.jag.2020.102154

Yamany, S. M., Farag, A. A., & Hsu, S.-Y. (1999). A fuzzy hyperspectral classifier for automatic target recognition (ATR) systems. En Pattern Recognition Letters (Vol. 20, Issues 11–13, pp. 1431–1438). Elsevier BV. https://doi.org/10.1016/s0167-8655(99)00116-6

Wang, D., Lu, X., & Rinaldo, A. (2017). DBSCAN: Optimal Rates For Density Based Clustering (Versión 3). arXiv. https://doi.org/10.48550/ARXIV.1706.03113

Jang, J., & Jiang, H. (2018). DBSCAN++: Towards fast and scalable density clustering (Versión 3). arXiv. https://doi.org/10.48550/ARXIV.1810.13105

Tran, T. N., Drab, K., & Daszykowski, M. (2013). Revised DBSCAN algorithm to cluster data with dense adjacent clusters. En Chemometrics and Intelligent Laboratory Systems (Vol. 120, pp. 92–96). Elsevier BV. https://doi.org/10.1016/j.chemolab.2012.11.006

Song, S., Zhou, H., Yang, Y., & Song, J. (2019). Hyperspectral Anomaly Detection via Convolutional Neural Network and Low Rank With Density-Based Clustering. En IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (Vol. 12, Issue 9, pp. 3637–3649). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/jstars.2019.2926130

Datta, A., Ghosh, S., & Ghosh, A. (2015). Combination of Clustering and Ranking Techniques for Unsupervised Band Selection of Hyperspectral Images. En IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (Vol. 8, Issue 6, pp. 2814–2823). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/jstars.2015.2428276

Windrim, L., Ramakrishnan, R., Melkumyan, A., Murphy, R. J., & Chlingaryan, A. (2019). Unsupervised Feature-Learning for Hyperspectral Data with Autoencoders. En Remote Sensing (Vol. 11, Issue 7, p. 864). MDPI AG. https://doi.org/10.3390/rs11070864

Mercier, G., & Lennon, M. (s. f.). Support vector machines for hyperspectral image classification with spectral-based kernels. En IGARSS 2003. 2003 IEEE International Geoscience

and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477). IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. IEEE. https://doi.org/10.1109/igarss.2003.1293752

Windrim, L., Ramakrishnan, R., Melkumyan, A., & Murphy, R. (2017). Hyperspectral CNN Classification with Limited Training Samples. En Procedings of the British Machine Vision Conference 2017. British Machine Vision Conference 2017. British Machine Vision Association. https://doi.org/10.5244/c.31.4

Schittenkopf, C., Deco, G., & Brauer, W. (1997). Two Strategies to Avoid Overfitting in Feedforward Networks. En Neural Networks (Vol. 10, Issue 3, pp. 505–516). Elsevier BV. https://doi.org/10.1016/s0893-6080(96)00086-x

Syakur, M. A., Khotimah, B. K., Rochman, E. M. S., & Satoto, B. D. (2018). Integration k-means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster. En IOP Conference Series: Materials Science and Engineering (Vol. 336, p. 012017). IOP Publishing. https://doi.org/10.1088/1757-899x/336/1/012017

McInnes, L., & Healy, J. (2017). Accelerated Hierarchical Density Based Clustering. En 2017 IEEE International Conference on Data Mining Workshops (ICDMW). 2017 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE. https://doi.org/10.1109/icdmw.2017.12

McInnes, L., Healy, J., & Astels S. (2016). HDBSCAN - PyPI. https://pypi.org/project/hdbscan

Anexos

Anexo 1: Eficacia de PCA

Para comprobar la eficacia de la reducción de dimensionalidad entregada por la técnica PCA se analizaron los datos de conjuntos de píxeles que representan a las cerezas, a las hojas y al cielo en la imagen del cultivo de cerezas Requínoa. La comparación está disponible en la figura 19.

Es posible observar cómo en cada una de las comparaciones los grupos de puntos que representan a cerezas y hojas son claramente diferenciables, lo que indica que el nivel de representatividad de los píxeles es la suficiente como para distinguir claramente distintas áreas potencialmente de interés dentro de la imagen.

Además, es posible determinar que el cielo es claramente un conjunto de píxeles más apartado respecto a las regiones orgánicas (cerezas y hojas) dentro de la imagen.

PCA

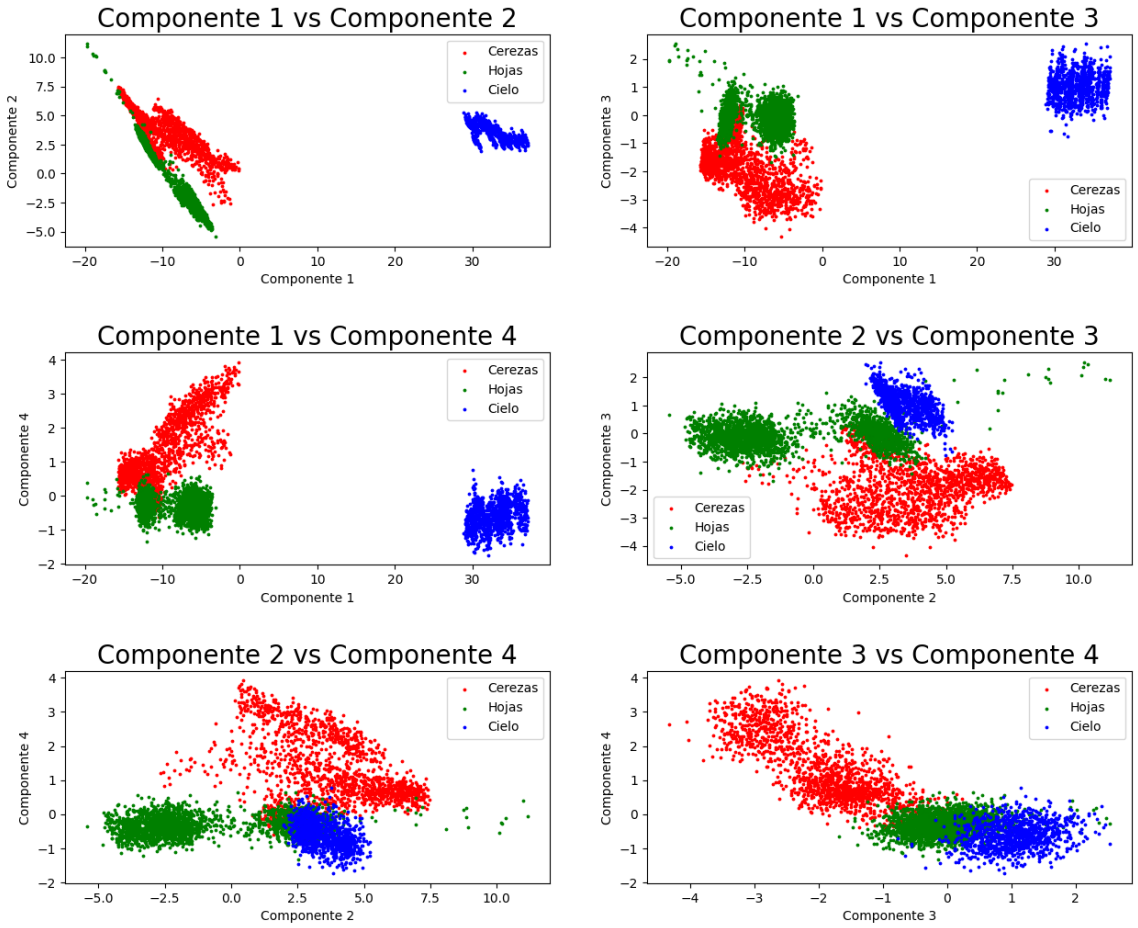

Figura 19. Primeras cuatro componentes principales comparadas entre sí.