Máster en Ingeniería de Sistemas y Control





Universidad Nacional de Eduación a Distancia
Universidad Complutense de Madrid

Trabajo Final de Máster

CLASIFICACIÓN DE TEXTURAS EN IMÁGENES AÉREAS

Alumno:
Antonio Vidal López

Director:
Gonzalo Pajares Martínsanz

28 de mayo de 2019

Autorización

El abajo firmante, matriculado en el Máster en Ingeniería de Sistemas y Control de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) y a la Universidad Nacional de Educación a Distancia (UNED) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado:

Antonio Vidal López

Agradecimientos

Me gustaría dar las gracias a mis padres, a mi hermana y a mi pareja por todo su apoyo y por recordarme cada día que lo más importante es disfrutar todo lo que uno hace. Del mismo modo, todo mi agradecimiento al personal docente de la Universidad Complutense de Madrid y Universidad Nacional de Educación a distancia, en especial a mi director de Trabajo Fin de Máster Gonzalo Pajares, por confiar en mí, por toda su dedicación y ayuda a lo largo del presente trabajo y durante la asignatura de Visión por Computador. Finalmente, a todos a mis compañeros de trabajo por sus ánimos y apoyo para la culminación del máster.

Resumen

El presente trabajo se encuadra en el área de Visión Artificial, en lo referente al tema de clasificación de texturas en imágenes aéreas. El objetivo es determinar el comportamiento de métodos de clasificación de texturas en dichas imágenes, evaluando dicho comportamiento mediante la utilización de índices que permiten establecer la calidad de las particiones que establecen dichos métodos.

El los últimos años, con el advenimiento de drones equipados con cámaras de visión operando en el espectro visible, surge la necesidad de identificar, analizar o extraer texturas de imágenes. Una textura es un elemento visual que posee cualidades ópticas derivadas de su estructura física, siendo este último aspecto muy importante, ya que la textura es un elemento visual que sensibiliza y caracteriza materialmente las superficies de los objetos o elementos de la imagen. Un ejemplo particular son las imágenes aéreas en las que se pueden observar claramente diferentes tipos de texturas terrestres caracterizadas por el color de la vegetación, terrenos, montañas o incluso zonas de agua como pueden ser mares o ríos. Es en el ámbito del análisis de imágenes aéreas y su caracterización en función de las propiedades espectrales derivadas de la textura donde se enfoca el presente trabajo.

Dada la necesidad de lo expuesto anteriormente, este trabajo realiza una evaluación de diversos algoritmos con la finalidad de satisfacer dichas necesidades. Se utilizan algoritmos basados en la teoría de redes neuronales, concretamente el algoritmo competitivo de Lloyd, la Cuantización Vectorial y el Self-Organizing Map, y otros basados en estimadores estadísticos, es el Fuzzy K-Means y el algoritmo paramétrico de Bayes.

El análisis se divide en dos fases: fase de aprendizaje y fase de clasificación, siendo en la fase de aprendizaje donde se realiza la comparativa de los algoritmos entre sí mediante una serie de índices de evaluación de la calidad de la partición, a saber: diámetro máximo del grupo, separación de *cluster* mínima, separación, cohesión, Ball-Hall, Hartigan, Xu, Calinski-Harabasz y coeficiente de Silhouette.

Finalmente, se muestran los resultados obtenidos durante la fase de clasificación sobre imágenes áreas, junto con un análisis y la extracción de sus texturas clasificadas.

Palabras clave: *Métodos de clasificación, partición, índice de validación, textura, reconocimiento de patrones, imágenes aéreas.*

Abstract

This work is framed in the area of Artificial Vision, regarding the subject of texture classification in aerial images. The goal is to determine the behavior of methods of classification of textures in such images, evaluating the behavior of such classifiers through the use of indices that allow to establish the quality of the partitions provided by these methods. In recent years, with the advent of drones equipped with vision cameras operating in the visible spectrum, there is a need to identify, analyze or extract textures of images. A texture is a visual element that has optical qualities derived from its physical structure, this latter aspect is very important, since the texture is a visual element that sensitizes and characterizes the object's surfaces based on the material or elements in the image. A particular example are the aerial images where different types of terrestrial textures can be clearly observed characterized by the color of vegetation, land, mountains or even water areas, as seas or rivers. This work is focused on the analysis of aerial images and its characterization, based on the spectral properties, derived from the texture.

According to the above, this work assesses several algorithms in order to determine their performance. With such purpose, algorithms based on the theory of neural networks are used, specifically the competitive algorithm of Lloyd, the Vector Quantization and the Self-Organizing Map, and others based on statistical estimators as the Fuzzy K-Means and the Bayes parametric approach.

The analysis is divided into two phases: learning phase and classification phase, being in the learning phase where the algorithms are compared to each other by means of different indices that measure the partition quality, namely: maximum diameter of the group, minimum cluster separation, separation, cohesion, Ball-Hall, Hartigan, Xu, Calinski-Harabasz and Silhouette coefficient.

Finally, the results obtained during the classification phase are shown, on aerial images, together with an analysis and extraction of their classified textures.

Keywords: *Clasification methods, partition, validation indices, textures, pattern recognition, aerial images.*

Índice general

Resumen **5**

Abstract **7**

1 Introducción **15**
1.1. Motivación 15
1.2. Objetivos 16
1.3. Metodología 17
1.4. Organización de la memoria 18

2 Fundamentos teóricos **19**
2.1. Aprendizaje Máquina 19
2.1.1. Reconocimiento de patrones 20
2.1.2. Redes Neuronales 21
2.1.3. Estimadores estadísticos 22
2.2. Teoría del color 23
2.3. Revisión de métodos de agrupamiento y clasificadores 24
2.3.1. Fuzzy K-Means 25
2.3.2. Cuantización Vectorial 27
2.3.3. Algoritmo competitivo de Lloyd 28
2.3.4. Mapas auto-organizados (Self-Organizing Map) 29
2.3.5. Clasificador paramétrico de Bayes 30
2.3.6. Distancias e índices de validación de agrupamientos 32
2.3.6.1. Distancia Euclídea 32
2.3.6.2. Distancia de Mahalanobis 33
2.3.6.3. Índices de validación de la partición 33

3 Diseño de la arquitectura para análisis de imágenes **37**
3.1. Introducción y manejo de la información 37
3.2. Diseño 38
3.3. Análisis y estudio 40
3.3.1. Fase de aprendizaje 40
3.3.1.1. Partición de datos 40
3.3.1.2. Cálculo de centros 42
3.3.2. Fase de clasificación 45

4 Resultados **47**
4.1. Descripción de las imágenes 47

4.2. Comportamiento de los índices de validación en fase de aprendizaje . . 48
4.2.1. Evaluación de particiones 48
4.2.2. Evaluación de centros de *cluster* 48
4.3. Fase de clasificación 51
4.3.1. Imagen del subgrupo A 52
4.3.2. Imagen del subgrupo B 54

5 Conclusiones y trabajo futuro **57**

Bibliografía **59**

A Extracción de texturas **63**
A.1. Imagen subgrupo A 63
A.2. Imagen subgrupo B 66

B Código utilizado **69**

Índice de figuras

1.1. Fases y etapas del trabajo 18

2.1. Esquema de flujo general de aprendizaje máquina 19
2.2. Imagen aérea con diferentes texturas. 21
2.3. Tetraedro de color RGB. *(Pajares y Cruz, 2011)* 24

3.1. Lista de píxeles 37
3.2. Escenario implementado 39
3.3. Evolución de SSW con el aumento de clases en Fuzzy. 41
3.4. Evolución de SSW con el umbral en CV. 41
3.5. Correspondencia umbral y total de clases creadas por CV. 41
3.6. Coeficiente de Silhouette para las clases de Fuzzy y CV. (a) Con cuatro grupos; (b) Con seis grupos. 42
3.7. Comportamiento de los índices SSW, SSB y CH con el incremento del peso exponencial en Fuzzy K-Means 43
3.8. Evolución indice Calinski-Harabasz para diferentes razones de aprendizaje en el algoritmo de Lloyd. 44

4.1. Índices de validación de *cluster* para los algoritmos Fuzzy K-Means y CV . 48
4.2. Centros en particiones Fuzzy K-Means 49
4.3. Índices de validación de *cluster* sobre particiones Fuzzy K-Means 50
4.4. Centros en particiones CV 50
4.5. Índices de validación de *cluster* sobre particiones CV 51
4.6. Índices de validación de *cluster* sobre particiones CV y Fuzzy K-Means . . 51
4.7. Imagen subgrupo A 52
4.8. Texturas de la imagen del subgrupo A clasificadas mediante diferentes algoritmos 52
4.9. Correspondencia colores clases y centros de particiones 53
4.10. Distancia entre píxel original y centro de la clase donde ha sido clasificado. 53
4.11. Píxeles cuya distancia supera el umbral definido 54
4.12. Imagen subgrupo B 54
4.13. Texturas de la imagen del subgrupo B clasificadas mediante diferentes algoritmos 55
4.14. Distancia entre píxel original y centro de la clase donde ha sido clasificado. 55
4.15. Píxeles cuya distancia supera el umbral definido 56

A.1. Texturas clase 1 imagen subgrupo A 63
A.2. Texturas clase 2 imagen subgrupo A 64
A.3. Texturas clase 3 imagen subgrupo A 64
A.4. Texturas clase 4 imagen subgrupo A 65

A.5. Texturas clase 1 imagen subgrupo B 66
A.6. Texturas clase 2 imagen subgrupo B 66
A.7. Texturas clase 3 imagen subgrupo B 67
A.8. Texturas clase 4 imagen subgrupo B 67

B.1. Jerarquía de directorios código fuente 69

Lista de acrónimos

BH *Ball-Hall*

CH *Calinski-Harabasz*

CMY *Cyan, Magenta, Yellow*

CV *Cuantización Vectorial*

FC *Fuzzy Clustering*

HT *Hartigan*

RGB *Red, Green, Blue*

RNA *Red Neuronal Artificial*

SOM *Self-Organizing Map*

SSB *Sum of Square Between*

SSW *Sum of Square Within*

CAPÍTULO 1

Introducción

El presente trabajo tiene como objetivo el estudio y comportamiento de métodos de clasificación en imágenes digitales aéreas. Exactamente se utilizan métodos ubicados en el ámbito de los estimadores estadísticos como son el Fuzzy K-Means y el clasificador paramétrico de Bayes, así como los encuadrados en el ámbito de las redes neuronales básicas como son la Cuantización Vectorial, el Algoritmo competitivo de Lloyd y los Mapas auto-organizados (Self-Organizing Map).

1.1. Motivación

A lo largo de la existencia de la vida humana la visión ha sido uno de los mecanismos sensoriales de percepción más importantes, por supuesto, para nada exclusivo ya que cualquier incapacidad visual no impide el desarrollo de otras actividades mentales. Para los humanos, el reconocimiento de objetos o patrones en imágenes en general es algo innato y no presenta ninguna dificultad en especial.

Con el nacimiento de los computadores surge la necesidad de la manipulación de imágenes digitales, inicialmente motivado por el interés de mejora de la calidad para su publicación en periódicos o codificación en diferentes niveles de gris con el fin de realizar su transmisión a través de redes. Más adelante, van surgiendo otras necesidades fundamentadas, principalmente, en dos áreas de aplicación: mejora de la calidad para la interpretación humana y procesamiento de los datos de la escena para la percepción de las máquinas de forma autónoma. Dicho de otra manera, en ambos casos se intenta dotar a las máquinas con un sistema de visión, es aquí cuando surge el concepto de Visión Artificial.

La visión artificial puede describirse como la disciplina científica que aglutina un conjunto de métodos para adquirir, procesar y analizar imágenes del mundo real con el fin de producir información que pueda ser tratada por un computador. El principal objetivo es dotar a la máquina de ojos para ver lo que ocurre y así poder tomar decisiones para automatizar procesos.

La visión artificial tiene muchas aplicaciones de gran importancia siendo la medicina uno de los primeros ámbitos de aplicación para la detección de anomalías en imágenes médicas (resonancias, imágenes de rayos X, ecografías, escáneres, etc). Este tipo de análisis suelen apoyarse en texturas para la identificación de regiones de la imagen con posibles problemas. Con el paso de los años, la visión artificial ha ido creciendo de manera exponencial en aplicaciones, llegando a estar presente en muchos dispositivos tanto en el ámbito industrial como en dispositivos cotidianos como tecnologías móviles. En la actualidad casi todos los dispositivos electrónicos que contienen sistemas de captura (comúnmente cámaras digitales) incorporan algoritmos de visión artificial que aportan conocimiento añadido al dispositivo. Por ejemplo, el reconocimiento facial para el desbloqueo de *smartphones*, la detección automática de rostros en fotos, la detección de sonrisas para realizar la captura, etc. Este auge ha hecho que los algoritmos de visión artificial hayan evolucionado tanto en términos de efectividad, como consumo computacional y tiempo de procesado.

Estas capacidades de decisión no solo se basan en la visión artificial como tal, sino que la máquina ha de tener un periodo de aprendizaje para poder llegar a tomar decisiones en procesos posteriores. Este aprendizaje automático se conoce también como Aprendizaje Máquina. El aprendizaje máquina es una rama de la inteligencia artificial cuyo principal objetivo es el desarrollo de técnicas que permiten dotar a la máquina de conocimiento para posteriormente poder tomar decisiones en base a esa inteligencia.

En el caso concreto del procesamiento de imágenes por computador, y con el advenimiento de drones equipados con cámaras de visión operando en el espectro visible, surge la necesidad de identificar, analizar o extraer texturas de imágenes. Una textura es un elemento visual que posee cualidades ópticas derivadas de su estructura física, siendo este último aspecto muy importante, ya que la textura es un elemento visual que sensibiliza y caracteriza materialmente las superficies de los objetos o elementos de la imagen. Un ejemplo particular son las imágenes aéreas en las que se pueden observar claramente diferentes tipos de texturas terrestres caracterizadas por el color de la vegetación, terrenos, montañas o incluso zonas de agua como pueden ser mares o ríos. Es en el ámbito del análisis de imágenes aéreas y su caracterización en función de las propiedades espectrales derivadas de la textura donde se enfoca el presente trabajo.

1.2. Objetivos

A continuación se presentan los objetivos de este trabajo junto con una pequeña descripción de cómo se enfocan de manera global desde el punto de vista del análisis en el trabajo.

El principal objetivo de este trabajo es realizar una evaluación de distintos algoritmos de aprendizaje y clasificación para la identificación de texturas en imágenes aéreas. A grandes rasgos se establecen los siguientes objetivos principales:

- Estudiar métodos de clasificación para identificación de texturas en imágenes aéreas.

- Desarrollar una herramienta que permita realizar un análisis comparativo de métodos de clasificación de texturas en dichas imágenes, mediante los índices de validación de las particiones.
- Estudiar índices de validación de las particiones en las clasificaciones.

En primer lugar, se comienza realizando un estudio detallado de métodos para la identificación de texturas en imágenes aéreas. Estos métodos utilizan la información de los píxeles, es decir, los valores RGB espectrales de cada píxel, para poder aprender de forma autónoma mediante el paradigma de **aprendizaje generalmente supervisado**.

Una vez identificados los algoritmos, se desarrolla una herramienta para poder realizar una comparativa entre ellos. Con tal propósito se han adaptado los siguientes algoritmos: Fuzzy K-Means, Cuantización Vectorial (CV), Bayes, Self-Organizing Maps (SOM) y Lloyd. Los algoritmos de Lloyd y Cuantización Vectorial se usan tanto para la fase de aprendizaje como en la fase de clasificación, siendo dichos algoritmos de naturaleza supervisada por lo que se ha realizado una adaptación para transformarlos en no supervisada.

Una vez implementados dichos algoritmos, se introducen y aplican una serie de indicadores que permitan medir, tras el proceso de aprendizaje, cómo de buenas son las particiones, y de esta manera poder comparar la eficiencia de los algoritmos en cuanto a la distribución de las muestras en las clases y la separabilidad de éstas.

1.3. Metodología

Una vez establecidos los objetivos a alcanzar, la metodología y etapas que se han definido para el desarrollo de los puntos descritos en la sección anterior se concreta en lo siguiente.

1. Definición de la problemática que se desea abordar, mediante el uso de algoritmos de aprendizaje, siendo éstos objeto de la investigación.
2. Planteamiento del alcance y objetivos de la investigación.
3. Diseño y desarrollo de las metodologías.
4. Realización de experimentos y obtención de resultados experimentales.
5. Análisis e interpretación de los datos.
6. Conclusión general.
7. Planteamiento de trabajo a futuro.

La figura 1.1 muestra un esquema básico en cuanto al desarrollo de la metodología propuesta.

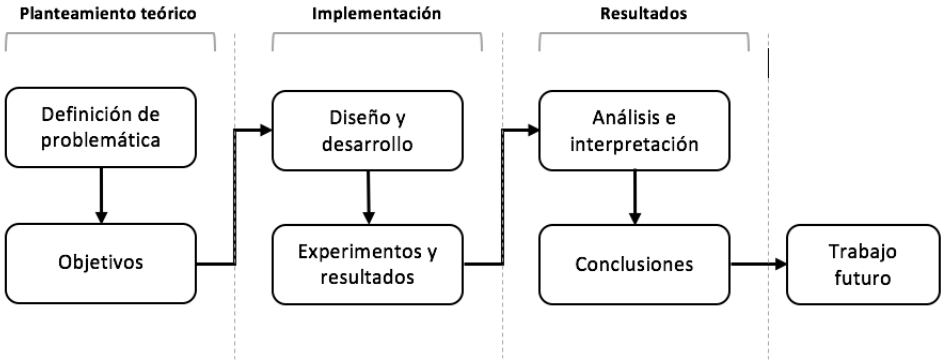


Figura 1.1: Fases y etapas del trabajo

Las etapas que componen la fase de planteamiento teórico, a saber definición de problemática y objetivos, se caracterizan por su enfoque teórico y de investigación, basándose fundamentalmente en la definición y estructuración del trabajo, así como la identificación de algoritmos y metodología a llevar a cabo para poder realizar un análisis en profundidad que permita comparar algoritmos de aprendizaje aplicados en imágenes aéreas para la clasificación a nivel de píxel.

Las etapas de implementación y resultados se centran en el diseño, desarrollo y análisis de resultados, por lo que se trata de la fase más técnica del trabajo, en la que se incluye la implementación de todas las herramientas necesarias para poder llevar a cabo los objetivos propuestos.

1.4. Organización de la memoria

El presente trabajo se organiza como sigue. Además del presente capítulo, contiene el capítulo dos, donde se introducen los conceptos teóricos que sustentan el trabajo. En el capítulo tres se describe el diseño y análisis de la aplicación desarrollada. El capítulo cuatro describe los resultados obtenidos, junto con su discusión. Finalmente, el capítulo cinco incluye las conclusiones y trabajo futuro.

CAPÍTULO 2

Fundamentos teóricos

En este capítulo se presenta una descripción teórica de las bases del trabajo, de los principales algoritmos y sus fundamentos.

2.1. Aprendizaje Máquina

El aprendizaje máquina, es una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las máquinas aprender, es decir, la creación de programas capaces de generalizar comportamientos a partir de un entrenamiento previo con ejemplos (Pajares y Cruz, 2011).

El proceso de aprendizaje comienza con observaciones o datos para buscar patrones en ellos y poder tomar decisiones futuras basados en dichos patrones, lo que permite a la máquina aprender automáticamente.

Como se puede observar en la figura 2.1 el proceso de aprendizaje automático consta principalmente de dos etapas: a) aprendizaje y b) decisión en base al conocimiento adquirido en el aprendizaje.

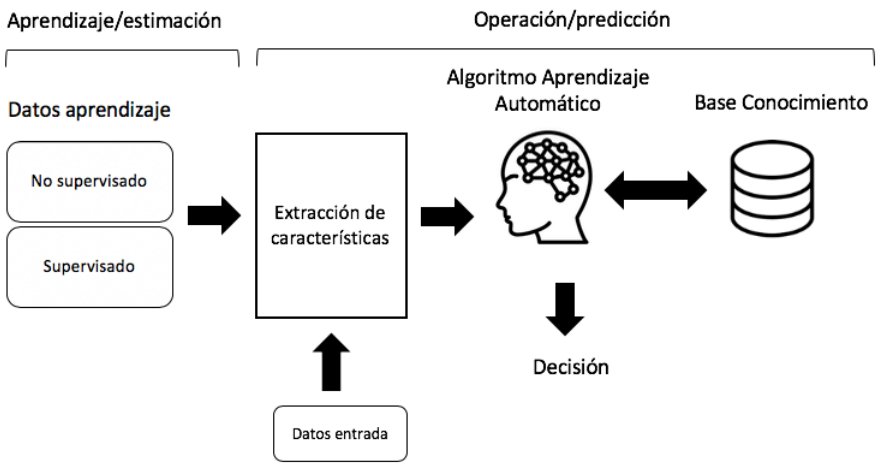


Figura 2.1: Esquema de flujo general de aprendizaje máquina

El proceso de aprendizaje se puede definir como la estimación de una dependencia desconocida o estructura de un sistema utilizando un número limitado de observaciones (datos o muestras de aprendizaje).

En la etapa de aprendizaje existen dos enfoques: supervisado y no supervisado.

Aprendizaje supervisado: Se utiliza para estimar una relación desconocida (entrada, salida) a partir de muestras conocidas (entrada, salida), es decir, los valores de salida para las muestras de entrenamiento son conocidos y por tanto, proporcionados para que el algoritmo de aprendizaje automático busque la relación existente entre las entradas y las salidas de las observaciones.

Matemáticamente se dice que el objetivo del aprendizaje supervisado es encontrar una función g, dado un conjunto de puntos de la forma $(x, g(x))$. Se supone que el conjunto de puntos conocidos que se usarán como entrenamiento son una muestra de variables aleatorias independientes idénticamente distribuidas que siguen generalmente una función de densidad de probabilidad conjunta.

Aprendizaje no supervisado: A diferencia del aprendizaje supervisado, en el aprendizaje no supervisado no se conoce el valor de salida, por lo que al sistema sólo se le proporcionan las muestras de entradas y no existe noción alguna de la salida durante la fase de aprendizaje. El objetivo del aprendizaje no supervisado puede ser estimar la distribución de probabilidad de las entradas o descubrir la estructura natural (agrupamientos o *clusters*) en los datos de entrada.

Para poder realizar la identificación de texturas de forma automática es necesario disponer de un conjunto de imágenes para entrenar el sistema y realizar el procedimiento de aprendizaje en base a esos datos de entrada.

2.1.1. Reconocimiento de patrones

El reconocimiento de patrones tiene como principal objetivo la clasificación de objetos en una serie de categorías o *clusters*. Estos objetos pueden estar contenidos en imágenes, ser parte de una señal o cualquier otro conjunto de datos. Nos referiremos a esos objetos de forma genérica utilizando el término *patrones* (Duda y col., 2012).

En este trabajo se utilizan patrones procedentes de imágenes aéreas, que poseen ciertas estructuras espaciales. Existen diferentes métodos para la descripción de elementos estructurales en la imagen para la identificación de las texturas, como matrices de co-ocurrencia u otros (Pajares y Cruz, 2007; Alegre y col., 2017). No obstante, esta caracterización espacial se manifiesta también bajo diferentes tonalidades espectrales, que es finalmente el enfoque en este trabajo, obviando los mencionados descriptores espaciales. La estructura de estos patrones puede no ser determinista, sino estadística, además puede no ser ni regular ni determinista, sino estadísticamemte regular. De forma más específica, los patrones que se utilizan son las componentes espectrales de cada

píxel en sus respectivos canales R (Rojo), G (verde) y B (azul).

En la figura 2.2 se pueden reconocer fácilmente las distintas texturas que la conforman, se observan regiones verdes, otras más áridas y algunas con vegetación intensa. Como se comenta en la sección anterior, para los humanos es fácil la distinción entre las diferentes texturas, pero para las máquinas es una ardua tarea.



Figura 2.2: Imagen aérea con diferentes texturas.

Tanto el aprendizaje supervisado como el no supervisado se basan en la extracción de patrones para encontrar la función que más se ajuste para representar las estructuras de entrada. El campo de reconocimiento de patrones tiene el objetivo de construir sistemas de reconocimiento de patrones artificiales que imiten las capacidades de reconocimiento del sistema humano. Para lograr este reconocimiento se hace uso de técnicas de aprendizaje automático. Para el aprendizaje se utilizan diferentes métodos que se pueden englobar en dos categorías: redes neuronales y aprendizaje estadístico.

El análisis llevado a cabo en este documento se basa en el aprendizaje automático sobre texturas caracterizadas por su naturaleza espectral en imágenes aéreas, en concreto, se realiza una comparativa entre diferentes algoritmos no supervisados encuadrados bajo el concepto de redes neuronales (Cuantización Vectorial, Self Organizing Map y Lloyd) y supervisados basados en aprendizaje estadístico (Fuzzy K-Means y Bayes).

2.1.2. Redes Neuronales

Una red neuronal artificial (RNA) es un modelo matemático inspirado en el comportamiento biológico de las neuronas. Una RNA está constituida por multitud de elementos más simples denominados neuronas y que se encuentran interconectados de una forma más o menos densa. Cada neurona está conectada con otras a través de unos enlaces, en los que el valor de salida de la neurona anterior es multiplicado por un valor de peso.

Una neurona desde el punto de vista biológico puede estar activa (excitada) o inactiva (no excitada), o lo que es lo mismo, se puede decir que tiene un estado de activación. En el caso de las neuronas artificiales también tienen diferentes estados de activación, algunas pueden tener solamente dos estados como las biológicas, si bien otras pueden tener varios estados dentro de un conjunto determinado. Esta activación viene establecida por una función conocida como "función de activación"que es la encargada de

calcular el estado de actividad de una neurona, transformando la entrada global en un valor de activación.

En el presente trabajo, los modelos de red neuronal utilizados no siguen exactamente el modelo de conectividad descrito, antes bien, las neuronas se caracterizan por lo que se puede identificar como los representantes de las clases, cuyos niveles de activación se establecen por la proximidad, en términos de distancias, entre las muestras y estos representantes, en el espacio de características y por tanto no existe conexión a través de pesos.

El uso de redes neuronales artificiales en procesos de aprendizaje tiene diversas ventajas:

- Autoaprendizaje: dado un conjunto de datos de entrada y la salida esperada es capaz de aprender y encontrar la función que mejor se ajusta a dicha salida.
- Son redes autoorganizadas, es decir, crean su propia representación de la información en su interior y lo hacen de forma transparente al usuario.
- Es tolerante a fallos ya que almacena la información de forma redundante.
- Es flexible ya que la respuesta del sistema no sufre cambios si hay cierto ruido o variaciones poco importantes en la información de entrada.
- Su estructura puede paralelizarse y ofrecer información en tiempo real.
- No linealidad, de forma general realiza una transformación no lineal de su entrada y la combinación de varias neuronas puede representar funciones no lineales.

2.1.3. Estimadores estadísticos

Un estimador de esta naturaleza se basa en una función estadística que permite obtener o identificar un parámetro desconocido de una población. Desde el punto de vista del agrupamiento mediante aprendizaje automático, se aplican estimadores sobre el conjunto de muestras de aprendizaje para posteriormente, en el caso de procesos de agrupamiento, poder utilizar sus valores en la fase de clasificación. Tanto el algoritmo Fuzzy K-Means como el clasificador de Bayes son de esta naturaleza.

Las propiedades de un estimador estadístico son las siguientes:

- **Sesgo:** Es la diferencia entre la esperanza (o valor esperado) del estimador y el verdadero valor del parámetro a estimar.
- **Eficiencia:** Un estimador es más eficiente o más preciso que otro estimador, si la varianza del primero es menor que la del segundo.
- **Consistencia:** A medida que el tamaño de la muestra crece, el valor del estimador debe tender a ser el valor del parámetro.

- **Robustez:** El no cumplimiento de los supuestos de partida en los que se basa la estimación no altera de manera significativa los resultados.
- **Suficiencia:** Un estimador es suficiente si resume toda la información relevante contenida en la muestra, de forma que ningún otro estimador pueda proporcionar información adicional sobre el parámetro desconocido de la población.
- **Invarianza:** Un estimador es invariante cuando el estimador de la función del parámetro coincide con la función del estimador del parámetro.

2.2. Teoría del color

El uso del color en el procesamiento de imágenes y aprendizaje máquina viene motivado principalmente porque permite la captura del contenido de la textura desde el punto de vista espectral, que caracteriza los patrones de aprendizaje utilizados en el presente trabajo, tal y como se ha descrito previamente.

Para distinguir un color de de otro, en el contexto de las imágenes digitales y desde el punto de vista de la descripción de las texturas, se utilizan las características de *brillo*, *matiz* y *saturación*.

- **Brillo**: modifica la forma tonal, es decir, incorpora la noción de intensidad de color.
- **Matiz**: se refiere a las diferencias cromáticas de colores. Así cabe distinguir entre verde bosque, verde pradera u otros tipos de verdes por ejemplo.
- **Saturación**: hace referencia a cómo un matiz aparece bajo ciertas condiciones de luz. Dicho de otro modo, se refiere a la pureza relativa o a la cantidad de luz blanca mezclada con un matiz.

Cuando se toman las características de matiz y saturación de forma conjunta se denominan *cromaticidad* y por tanto un color puede caracterizarse por su brillo y cromaticidad.

Con el propósito de facilitar la especificación de los colores, existen *modelos de color* estandarizados los cuales se basan principalmente en un sistema de coordenandas 3-D y un subespacio dentro de dicho sistema donde cada color es representado por un punto.

El modelo RGB (rojo, verde, azul) es el más común y es el utilizado en este estudio. En dicho modelo cada color es identificado por sus componentes primarias: rojo, verde y azul. El subespacio de color se representa en la figura 2.3.

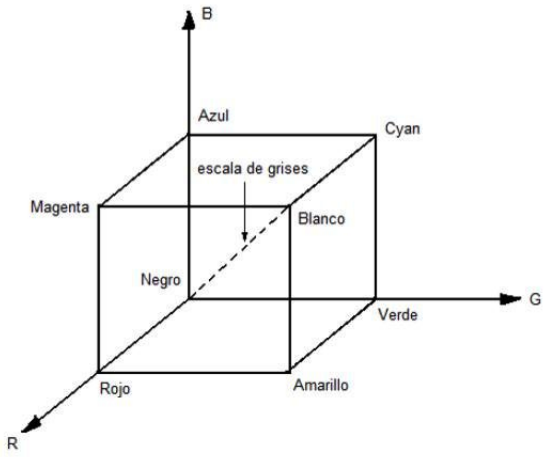


Figura 2.3: Tetraedro de color RGB. *(Pajares y Cruz, 2011)*

Como se observa en la figura 2.3 los valores RGB se sitúan en tres vértices y los colores *cyan*, *magenta* y *amarillo* se sitúan en otros tres. El negro se ubica en el origen de coordenadas, opuesto al blanco que es el punto más alejado del origen. Por convención se normalizan los valores RGB al rango $[0, 1]$, si bien es frecuente encontrarlos en el rango $[0, 255]$.

La principal aplicación, y la más importante, es la capacidad de que un punto pueda variar su color y su intensidad a través de los valores de sus componentes principales (RGB), además es un modelo muy eficiente desde el punto de vista de hardware, de hecho está presente en monitores y una amplia gama de cámaras de vídeo.

2.3. Revisión de métodos de agrupamiento y clasificadores

Los algoritmos de agrupamiento o *clustering* tienen como objetivo separar un conjunto de datos en un cierto número de grupos basándose en alguna medida de similitud. La finalidad es encontrar un conjunto de grupos para los cuales las muestras dentro de cada grupo están cohesionadas. Los algoritmos de agrupamiento se pueden dividir en las siguientes categorías:

- *Secuenciales:* producen agrupamientos simples. En la mayoría de los casos los vectores de características se presentan al algoritmo una sola vez, por tanto el resultado final tiende a depender del orden en el que se introducen los vectores al algoritmo.
- *Jerárquicos:* a su vez pueden ser de *aglomeración* o *división*. En aglomeración se parte de que cada dato es un único grupo y en cada iteración se van fusionando grupos. Por el contrario, en la división se parte de un único grupo conformado por todos los elementos del conjunto de datos.

- *Basados en la optimización de una función de coste:* Se utiliza una función de coste J para evaluar los agrupamientos. Por lo general el número de grupos se mantiene constante.

2.3.1. Fuzzy K-Means

El agrupamiento difuso (*Fuzzy* en inglés) es un tipo de agrupamiento donde cada elemento tiene un grado de pertenencia difuso a cada clase. El objetivo es dividir n elementos $x \in X$ que son caracterizados por p propiedades en k *clusters* o grupos (Duda y col., 2012; Alegre y col., 2016; Zimmermann, 1991; Bezdek y col., 1987).

Considerando el conjunto de datos $X = \{x_1, x_2, ..., x_n\} \in R^p$ que se desea clasificar en k clases $S_i = \{i = 1, .., k\}$ siendo S_i el centro del *cluster* i representado por el vector de características $v_i = \{v_{i1}, v_{i2}, ..., v_{ip}\} \in R^p$. Cada elemento x_i tendrá un grado de pertenencia u_{ik} a la clase k. Deben cumplirse las siguientes propiedades:

1. $u_{ik} \in [0, 1]$ para agrupamiento crisp o $u_{ik} \in [0, 1]$ para agrupamiento fuzzy, dado $1 \leq i \leq c; 1 \leq k \leq n$.
2. $\sum_{i=1}^{c} u_{ik} = 1$
3. $0 < \sum_{k=1}^{n} u_{ik} < n$

En la primera propiedad se indican dos casos: *crisp* y *fuzzy*. Como se ha comentado, en fuzzy cada elemento tiene un grado de pertenencia a cada clase, pero puede ser necesario agrupar de forma exclusiva un elemento a una clase en concreto, en ese caso la matriz de pertenencia tendrá valores 0 ó 1, lo que se conoce como agrupamiento *crisp*. A continuación se muestra un ejemplo:

$$U_{fuzzy} = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0{,}3 & 0{,}6 & 0{,}1 \\ 0{,}7 & 0{,}4 & 0{,}9 \end{bmatrix} \longrightarrow U_{crisp} = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Desde el punto de vista funcional, se trata de un algoritmo iterativo basado en minimizar una función objetivo, por lo que hay que definir un criterio de agrupamiento en la forma de una función objetivo que depende de la partición difusa. El objetivo es minimizar iterativamente esta función hasta obtener la partición difusa óptima. Para este estudio se utiliza la función de error mínimo cuadrático como función objetivo.

$$min\ z_m(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m \parallel \boldsymbol{x_k} - \boldsymbol{v_i} \parallel_G^2 \tag{2.1}$$

siendo

$$\| x_k - v_i \|_G^2 = (x_k - v_i)^t G (x_k - v_i) \tag{2.2}$$

Donde:

- $(x_1, x_2, ..., x_3) \in X$ son los datos.
- $v_i = (v_{i1}, v_{i2}, ..., v_{ic})$ es el vector centro del grupo i.
- $\| \ \|_G$ es la norma inducida por G.
- G es la matriz de pesos de dimensión (nxn). Si G es la matriz identidad, la norma inducida es la distancia euclidiana.

El peso asociado a cada distancia cuadrada u_{ik}^m es la m-ésima potencia del grado de pertenencia del k-ésimo dato al grupo i que se denomina *Peso Exponencial o difusor (fuzzifier)* y es, junto con el número de clases, un parámetro requerido por el algoritmo y cuyo principal objetivo es disminuir la influencia del ruido al obtener los centros de los *clusters*, reduciendo la influencia de los valores pequeños de u_{ik} frente a valores altos de u_{ik}. Cuanto mayor sea $m > 1$ mayor es dicha influencia.

Teniendo en cuenta lo descrito anteriormente, el procedimiento general del algoritmo K-Means se formaliza de la siguiente manera:

1. Inicializar los centros o pesos de forma aleatoria. En caso de establecer los centros aleatoriamente, se calculan los pesos a partir de dichos centros. Por el contrario, en caso de inicializar los pesos de forma aleatoria, se calculan los centros a partir de dichos pesos.
2. Inicializar la matriz de partición U
3. Calcular los centros de los grupos

$$v_i = \frac{1}{\sum_{k=1}^{n} (u_{ik})^m} \sum_{k=1}^{n} (u_{ik})^m x_k \qquad i = 1, ..., c \tag{2.3}$$

4. Actualizar la matriz de partición difusa U

$$u_{ik} = \frac{\left(\frac{1}{\|x_k - v_i\|_G^2} \right)^{2/(m-1)}}{\sum_{j=1}^{c} \left(\frac{1}{\|x_k - v_j\|_G^2} \right)^{2/(m-1)}} \qquad i = 1, ..., c; \quad k = 1, ..., n \tag{2.4}$$

5. Comprobar criterio de terminación

$$\Delta = \| U^{t+1} - U^t \|_G \tag{2.5}$$

Si $\Delta > \varepsilon$, siendo ε el criterio de terminación establecido, en caso contrario volver al paso 3.

Para el presente trabajo se ha realizado una adaptación del algoritmo Fuzzy K-Means para la inicialización de los centros de forma pseudo-aleatoria basada en Balasko y col. (2006), dado por la siguiente expresión y teniendo en cuenta que el número de *clusters* es dos.

$$\boldsymbol{v} = 2D\overline{\boldsymbol{M}} \circ \boldsymbol{R} + D\overline{\boldsymbol{m}} \tag{2.6}$$

donde $\overline{\boldsymbol{m}}$ es la media de los valores del conjunto de muestras disponibles $\boldsymbol{X}$, previamente normalizados $\boldsymbol{Y}$, $\overline{\boldsymbol{M}} = max(abs(\boldsymbol{Y} - \overline{\boldsymbol{m}}))$ es una matriz de dimensión $1 \times p$, $D = \left[\underbrace{1 \cdots 1}_{c}\right]^t$ cuya dimensión es $k \times 1$; $\boldsymbol{R}$ es una matriz de números aleatorios de dimensión $c \times p$ donde cada elemento de la matriz se obtiene utilizando la función $(rand(.) - 0{,}5)$, los valores aleatorios están restringidos al rango $[0, 1]$.

Para realizar la normalización de las muestras en el rango $[0, 1]$ se utiliza la siguiente ecuación:

$$Y = \frac{X - MIN}{MAX - MIN} \quad MIN = min\{X\}; \quad MAX = max\{X\} \tag{2.7}$$

2.3.2. Cuantización Vectorial

A diferencia del agrupamiento *Fuzzy*, la agrupación mediante Cuantización vectorial (CV) es exclusiva, por lo que un elemento x pertenecerá íntegramente a una clase o grupo cuyos elementos estén cohesionados. Otra diferencia radica en el número de clases ya que para CV no es conocido a priori, por lo que no solo agrupa los elementos en grupos cohesionados sino que establece el número de clases en los que realizar los agrupamientos. Previamente a la descripción del funcionamiento del algoritmo es importante conocer el concepto de cuantificador vectorial (Cherkassky y Mulier, 1998; Bishop, 1995; Patterson, 1996).

Un cuantificador vectorial Q es una proyección del espacio euclídeo d-dimensional $\Re^d$ donde $d \geq 2$ en un subconjunto finito C de $\Re^d$. Por tanto,

$$Q : \Re^d \rightarrow C \tag{2.8}$$

A partir de esta definición, las regiones que definen la partición son disjuntas y su unión es $\Re^d$. Un cuantizador se puede definir de una forma unívoca definiendo el conjunto de salida C y la correspondiente partición $\{R_j\}$ (Pajares y Cruz, 2011).

Existen varias condiciones necesarias para que un cuantificador vectorial sea óptimo, son las condiciones de Lloyd-Max (Lloyd 1957, Max 1970):

1. Vectores de salida: los vectores de salida deben ser el centroide de X y suponiendo que $X \in R_j$.

2. Cuantización óptima para las regiones: las partición debe satisfacer $R_j \supset \{x \in \Re^d : \| x - c_j \| < \| x - c_k \| \; \forall k \neq j\}$.

Dado que el número de clases no se conoce inicialmente, el algoritmo comienza suponiendo que hay una única clase y un umbral de referencia para la distancia de separación máxima permitida entre los datos de una misma clase, este umbral se define como T.

1. Para cada patrón se calcula la distancia con todos los centros existentes. Para el primer elemento él mismo constituye el primer centro.
2. Se toma el centro más cercano utilizando una medida de distancia, por ejemplo la distancia euclídea.
3. Si la distancia es menor que el umbral T, se asocia el elemento a la clase y se calcula la media de todos los elementos que pertenecen a dicha clase. Esta media nos proporciona el nuevo centro del grupo.
4. Si la distancia es mayor que el umbral T se crea una nueva clase, asignando el valor del centroide al del elemento.

2.3.3. Algoritmo competitivo de Lloyd

El algoritmo competitivo de Lloyd es una versión modificada del Algoritmo Generalizado de Lloyd (AGL). El AGL fue propuesto inicialmente por Lloyd (1975) y generalizado más tarde por Linde y col (1980). Este algoritmo aplica las dos condiciones necesarias que hemos expuesto anteriormente para determinar cuantizadores vectoriales óptimos minimizando el riesgo empírico.

El procedimiento es el siguiente:

1. **Inicio**: dados los puntos de datos $x(k), k = 1, 2, ...$ y centros de salida iniciales $c_j(0), j = 1, ..., m$.
2. Determinar el centro $c_j(k)$ más próximo al punto $x(k)$ (competición).

$$j = argmin \parallel x(k) - c_j(k) \parallel \tag{2.9}$$

3. Actualizar el centro de salida utilizando las ecuaciones:

$$c_j(k_j + 1) = c_j(k_j) + \gamma(k_j)\lfloor x(k) - c_j(k_j) \rfloor \tag{2.10}$$

$$k_j = k_j + 1 \tag{2.11}$$

donde $\gamma(k_j)$ es la razón de aprendizaje.

Para el cálculo de los centros iniciales se suele realizar la media aritmética de los elementos que pertenecen a dicha clase.

2.3.4. Mapas auto-organizados (Self-Organizing Map)

Un mapa auto-organizado (*Selg-Organizing Map* en inglés) es un tipo de red neuronal artificial que es entrenada mediante aprendizaje no supervisado siendo principalmente utilizada para la reducción de la dimensionalidad de los datos. Los mapas auto-organizados difieren de otras redes neuronales en el sentido de que éstos usan una función de vecindad para preservar las propiedades topológicas del espacio de entrada (Kohonen 1989, 1995).

Cuando se habla de reducir la dimensionalidad se hace referencia a encontrar una proyección de un espacio de las muestras de entrada $\Re^d$ en un espacio de salida Re^m donde $m < d$:

$$G(x) : \Re^d \rightarrow \Re^m \tag{2.12}$$

Este espacio de salida $\Re^m$ puede discretizarse en un conjunto de valores llamado mapa (*map*). Un requisito importante de este conjunto es que debe existir una distancia entre los miembros del mapa.

Desde el punto de vista estructural, un mapa auto-organizado está compuesto por varios nodos o neuronas. Cada neurona tiene asociado un vector de pesos de la misma dimensión que los vectores de entrada, además de una posición en el mapa. Para el mapa se utiliza generalmente un conjunto regular de puntos igualmente espaciados formando una rejilla que puede ser hexagonal o rectangular.

Dado que se trata de una red de aprendizaje competitivo, en general este tipo de redes están formadas por dos capas de neuronas:

- *Capa de entrada:* recibe los datos y mide la similitud ente esta entrada y los pesos almacenados en la red.
- *Capa de salida:* tiene neuronas competitivas que tienen conexiones inhibidoras entre ellas.

Inicialmente todos los pesos se establecen de forma aleatoria para que, cuando se introduce un patrón a la red en el inicio del proceso de aprendizaje y éste se propaga de la capa de entrada a la de salida, las neuronas, en base a sus pesos y función de activación se calcula la similitud del patrón, quedando activa la neurona ganadora, es decir, la que más se aproxime al valor de entrada. Para medir la similitud se pueden utilizar diferentes distancias, como la distancia Euclídea o de Mahalanobis (2.3.6).

En términos de aproximación estocástica (Pajares y Cruz, 2011), el algoritmo SOM se desarrolla de la siguiente forma:

Condiciones de partida: Dado un espacio de características discreto $\Psi = \{\Psi_1, \Psi_2, .., \Psi_b\}$, los puntos $x(k)$ y las unidades $c_i(k)$, con $j = 1, ..., b$ con el índice discreto k.

1. Determinar la unidad más próxima al dato $\mathbf{x}(k)$, que se conoce como la unidad vencedora:

$$\mathbf{z}(k) = \Psi\Big(\arg_j min \parallel c_j(k-1) - \mathbf{x}(k) \parallel^2 \Big) \tag{2.13}$$

2. Actualizar todas las unidades utilizando la ecuación de actualización:

$$\begin{aligned} &\mathbf{c}(k) = \mathbf{c}_j(k-1) + \gamma(k) K_{\alpha(k)}\Big(\Psi(j), \mathbf{z}(k)\Big)\Big[\mathbf{x}(k) - \mathbf{c}_j(k-1)\Big] \\ &k_j = k_j + 1 \end{aligned} \tag{2.14}$$

   con $j = 1, ..., b$.

3. Decrementar la razón de aprendizaje $\gamma(k)$ y el ancho de la vecindad $\alpha(k)$. K_α es una *función de vecindad* con un parámetro $\alpha(k)$ que define el ancho de la función o zona de influencia de la misma y que a su vez se le permite que varíe con k.

Una de las peculiaridades de este algoritmo es el decrecimiento gradual del ancho de la función de vecindad a medida que progresan las iteraciones. Usualmente se suelen utilizar las siguientes ecuaciones tanto para K_α como para $\alpha(k)$:

$$K_{\alpha(k)} = (\mathbf{z}, \mathbf{z'}) = exp\left(\frac{\parallel \mathbf{z} - \mathbf{z'} \parallel^2}{2\alpha^2(k)} \right) \; con \; \alpha(k) = \alpha_{inicial} \left(\frac{\alpha_{final}}{\alpha_{inicial}} \right)^{k/k_{max}} \tag{2.15}$$

donde k es el número de iteración y k_{max} el número máximo de iteraciones. El ancho inicial de la vecindad $\alpha_{inicial}$ se elige de modo que la vecindad cubra todas las unidades y el acho final α_{final} controla el grado de variación del núcleo entre iteraciones consecutivas, siendo estos parámetros fijos una vez establecidos.

2.3.5. Clasificador paramétrico de Bayes

El método basado en la teoría de la decisión de Bayes es un método estadístico clásico en clasificación de patrones. Se fundamenta en el supuesto de que el problema de la decisión se enfoca en términos probabilísticos y que todas las probabilidades relevantes resultan conocidas (Duda y col., 2012).

El aprendizaje de este algoritmo consiste en el cálculo de medias y matrices de covarianza para cada una de las particiones obtenidas mediante algún otro algoritmo como puede ser Fuzzy K-Means o Cuantización Vectorial.

Dadas un total de c clases o particiones, para cada clase la media se calcula de la siguiente forma:

$$m_{c_i} = \frac{1}{n} \sum_{j=1}^{n} x_j \tag{2.16}$$

donde n es el total de muestras de la partición c_i y x el vector características de cada muestra.

En el caso de la covarianza, el cálculo se realiza aplicando la ecuación (2.17).

$$C_{c_i} = \frac{1}{n-1} \sum_{j=1}^{n} (x_j - m)(x_j - m)^t \tag{2.17}$$

Para la clasificación de una muestra se utiliza la probabilidad de pertenecer a cada una de las clases. Dicha probabilidad viene dada por la distancia de Mahalanobis, la cual es calculada en base a las medias y matriz de covarianza obtenidas en la fase de aprendizaje.

Sea y el estado que designa la pertenencia a una de las clases y considerando y como una variable aleatoria, ya que la pertenencia a un estado es impredecible, se puede establecer que existe alguna *probabilidad a priori* $P(y = c_i)$ de que la muestra pertenezca a la clase c_i siendo $c_i \in c$ donde c es el total de clases. Todas las probabilidades son no negativas y su suma es la unidad.

Para decidir en que clase ha de clasificarse la muestra tratada, la única información disponible son las probabilidades calculadas, por tanto parece razonable utilizar la siguiente *regla de decisión*: establecer la clase con mayor probabilidad. No obstante, dado que una muestra puede tener más de una característica, se puede construir un vector de características $x = \{x_1, x_2, ..., x_n\}$, así pues diferentes muestras poseerán diferentes valores en x, por lo que resulta natural expresar esta variabilidad en términos probabilísticos. En ese sentido, se considera x una variable aleatoria continua cuya distribución depende del tipo de clase.

Sean $p(x|y = c_i)$ las funciones de *densidad de probabilidad condicionales* para x dado que la muestra pertenezca a la clase c_i. Supongamos que se conocen tanto las probabilidades a priori como estas últimas funciones de densidad de probabilidad, mediante la *regla de Bayes* podemos determinar la influencia de cada características con la pertenencia a cada clase:

$$p(y = c_i|x) = \frac{p(x|y = c_i)P(y = c_i)}{p(x)} \tag{2.18}$$

donde

$$p(x) = \sum_{i=1}^{c} p(x|y = c_i)P(y = c_i) \tag{2.19}$$

La regla de Bayes muestra cómo la observación del valor x cambia las *probabilidades a priori* a las *probabilidades a posteriori* $p(y = c_i|x)$.

La ecuación (2.19) es una constante de normalización, que no necesita ser obtenida a la hora de tomar una decisión, puesto que la regla de decisión es una comparación de las magnitudes relativas de las probabilidades a posteriori. Una vez determinadas dichas probabilidades a posteriori, la siguiente regla (función discriminante) se utiliza para clasificar x:

$$x \in c_i \ si \ P(y = c_i|x) > P(y = c_j x) \ \ \forall i \neq j \ \ i, j = 1, ..., c \tag{2.20}$$

Fijándose en el segundo término de la expresión (2.18) del teorema de Bayes y eliminando el término no discriminante $p(x)$, se obtiene una forma alternativa de clasificar un vector de características x:

$$x \in c_i \ \ sii \ \ d_M^2(x, m_i) < d_M^2(x, m_j) \ \ \forall i \neq j, \ \ i, j = 1, 2, ..c \tag{2.21}$$

donde m_i y m_j son los vectores media de las clases c_i y c_j respectivamente, d_M es la distancia de Mahalanobis.

La ecuación (2.21) se aplica para distribuciones de densidad de probabilidad normales o Gaussianas y distribuciones cuya probabilidad inicial es igual para todas las clases.

2.3.6. Distancias e índices de validación de agrupamientos

Para la validación de particiones obtenidas mediante los algoritmos de clasificación es necesario hacer uso de un conjunto de índices que permitan medir cómo de buenas son dichas particiones, además de servir como variables de comparación para los resultados obtenidos por cada algoritmo.

Antes de analizar los diferentes índices implementados en el presente estudio, es importante mencionar que la mayoría de ellos se basan en métricas de distancias para realizar el cálculo: distancia de Mahalanobis y distancia Euclídea.

2.3.6.1. Distancia Euclídea

La distancia Euclídea es una función no negativa usada en diversos contextos para calcular la distancia entre dos puntos en un espacio de n dimensiones. En general la distancia euclidiana entre los puntos $A = (a_1, a_2, ..., a_n)$ y $B = (b_1, b_2, ..., b_n)$ del espacio euclídeo n-dimensional, se define como:

$$d(A, B) = \sqrt{(a_1 - b_1) + (a_2 - b_2) + ... + (a_n - b_n)} = \sqrt{\sum_{i=1}^{n} (a_i - {b_i}^2)} \tag{2.22}$$

2.3.6.2. Distancia de Mahalanobis

La distancia de Mahalanobis es una medida de distancia introducida por Mahalanobis en 1936 (Duda y col., 2012). Su principal utilidad es determinar la similitud entre dos variables aleatorias multidimensionales. A diferencia de la distancia Euclídea, se tiene en cuenta la correlación entre variables aleatorias.

Formalmente, la distancia de Mahalanobis entre dos variables aleatorias con la misma distribución de probabilidad $\vec{x}$ y $\vec{y}$ con matriz de covarianza $\sum^{-1}$ se define como:

$$d_m = (\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \sum^{-1} (\vec{x} - \vec{y})} \tag{2.23}$$

La distancia de Mahalanobis cumple las siguientes propiedades propias de toda distancia:

- **Semipositividad**: $d(a, b) \geq 0 \ \ \forall a, b \in X$ y además $d(a, b) = 0$ si $a = b$
- **Simetricidad**: $d(a, b) = d(b, a) \ \ \forall a, b \in X$
- **Desigualdad triangular**: $d(a, b) \leq d(a, c) + d(c, b) \ \ \forall a, b, c \in X$

2.3.6.3. Índices de validación de la partición

A continuación se describen los diversos índices de validación de las particiones obtenidas mediante las funciones utilizadas en el presente estudio y que tienen como objetivo servir de comparativa entre los resultados obtenidos por los algoritmos de clasificación descritos previamente para las clasificación de las texturas.

La validación puede ser **externa** o **interna**, la principal diferencia radica en el uso o no de información externa durante el proceso de validación, es decir, información que no es fruto de la técnica de agrupación utilizada. Las técnicas de validación interna, a diferencia de las externas, miden la calidad de la agrupación basándose únicamente en la información de los datos. Evalúan cómo de buena es la el resultado del agrupamiento sin necesidad de información ajena al propio algoritmo y su resultado.

Para este estudio se utilizan los índices de validación internos descritos a continuación (Balasko y col., 2006; Bezdek, 1981; Bensaid y col., 1996; Xie y Beni, 1991; Dunn, 1976; Kim y col., 2003).

Diámetro Máximo del grupo (Δ)

Se define como **Diámetro Máximo del Grupo** a la máxima distancia entre cualesquiera dos puntos pertenecientes al mismo grupo, es decir, el mayor diámetro entre todos los puntos que pertenecen a una clase C. Se calcula aplicando la ecuación (2.24).

$$\Delta = max(\Delta_i) \ \ \forall \ i \in C; \quad \Delta_i = max \ d(j, j') \ \ \forall \ j \in i \tag{2.24}$$

El diámetro máximo permite evaluar la compacidad de la partición, tomando valores en el rango $[0, 1)$ y cuyo objetivo es minimizar dicho valor. La compacidad es la propiedad por la que los elementos de una misma agrupación deben estar tan cerca como sea posible.

Separación de *Cluster* Mínima (δ)

Se define como **Separación de *Cluster* Mínima** la menor distancia entre cualquiera de dos objetos pertenecientes a agrupaciones diferentes, es decir, se trata de la menor distancia entre todas las agrupaciones. Se calcula mediante la ecuación (2.25).

$$\delta = min\ \delta(c, c') \quad c, c' \in C; \quad \delta(c, c') = min\ d(i, i') \quad i \in c, i' \in c' \tag{2.25}$$

El rango de valores obtenidos se encuentra entre $[0, 1)$ y el objetivo es maximizarlo.

Separación *(Sum Square Between)*

Las agrupaciones deben estar ampliamente separadas entre sí, es decir, deben estar lo suficientemente separadas unas de otras. Para el cálculo de la separación se aplica la ecuación (2.26).

$$SSB = \sum_{i=1}^{k} n_i\ dist^2\ (c_i - \overline{x}) \tag{2.26}$$

Siendo k el número de agrupaciones, n_i el número de elementos en la agrupación i, c_i el centroide de la agrupación i y $\overline{x}$ la media de todo el conjunto de datos.

El objetivo es maximizar el valor de SSB (*Sum Square Between*) con la finalidad de obtener agrupaciones lo más separadas posible entre sí.

Cohesión *(Sum Square Within)*

Un elemento de cada agrupación debe ser lo más cercano posible a otros miembros de la misma agrupación. Para medir la cohesión se aplica la ecuación (2.27).

$$SSW = \sum_{i=1}^{k} \sum_{x \in C_i} dist^2\ (m_i, x) \tag{2.27}$$

Siendo k el número agrupaciones, x un punto de la agrupación C_i y m_i el centroide de la agrupación C_i.

El objetivo es minimizar el valor de SSW (*Sum Square Within*) con la finalidad de obtener agrupaciones lo más cohesionadas posibles.

Ball-Hall (BH)

El índice de **Ball-Hall** permite medir la dispersión media por agrupación. Para calcularlo se aplica la ecuación (2.28).

$$BH = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n_i}\sum_{x\in C_i} dist^2\ (m_i, x) \tag{2.28}$$

donde k es el total de agrupaciones, n_i el número de elementos de la agrupación i, x un elemento de la clase i y m_i el centroide de la clase i.

Dado que el indicador mide la dispersión media por agrupación, el objetivo es minimizar su valor.

Hartigan (HT)

El índice de **Hartigan** permite relacionar los índices de separación (SSB) y cohesión (SSW). Se calcula mediante la ecuación (2.29).

$$HT = log\left(\frac{SSB}{SSW}\right) \tag{2.29}$$

Dado que el objetivo es maximizar la separación (SSB) y minimizar la cohesión (SSW), cuanto mayor es el índice de HT mejor es el agrupamiento obtenido. Por tanto, el objetivo es maximizarlo.

Xu

El índice de **Xu** permite relacionar la cohesión con el total de elementos, teniendo en cuenta la dimensión de los mismos. Se calcula mediante la ecuación (2.30).

$$XU = d \cdot log\left(\sqrt{\frac{SSW}{dN^2}}\right) + log(k) \tag{2.30}$$

Siendo d la dimensión de los datos, SSW la cohesión, N el número de elementos y k el total de agrupaciones.

El objetivo es minimizar su valor, a menor valor más cohesionadas estarán las agrupaciones.

Calinski-Harabasz (CH)

El índice de **Calinski-Harabasz** se basa en el concepto de agrupaciones cohesionadas y bien separadas. Para su cálculo se aplica la ecuación (2.31).

$$CH = \frac{SSB/(k-1)}{SSW/(n-k)} \tag{2.31}$$

Donde SSB es la separación, k el total de agrupaciones, SSW la cohesión y n el total de elementos.

El objetivo es maximizar su valor.

Coeficiente de Silhouette

El coeficiente de Silhouette permite evaluar cómo de similar es un objeto al *cluster* a que pertenece (cohesión) comparado con otros *clusters* (separación). La forma más común de uso es mediante gráficas que permiten evaluar visualmente la consistencia de los *clusters*, siendo un *cluster* consistente aquel cuyos elementos están cohesionados y bien separados. El rango de valores que puede tomar está en $[-1, 1]$ donde valores valores cercanos a 1 indican que un elemento es consistente en el *cluster* asignado.

Para cada cada elemento $i \in C_i$ (elemento i perteneciente al *cluster* C_i):

$$a(i) = \frac{1}{\mid C_i \mid -1} \sum_{j \in C_i, i \neq j} d(i,j) \tag{2.32}$$

Siendo $a(i)$ la distancia media entre el elemento i y todos le elementos de su *cluster*. $d(i,j)$ la distancia entre los elementos i y j en el *cluster* C_i. La distancia puede ser la Euclídea o Manhattan.

A continuación se define la similitud entre el elemento i al *cluster* c como la media de las distnacias de i a todos los elementos en c.

$$b(i) = \min_{i \neq j} \frac{1}{\mid C_j \mid} \sum_{j \in C_j} d(i,j) \tag{2.33}$$

Finalmente, se define el coeficiente de Silhouette como:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}, \quad si \quad \mid C_i \mid > 1 \tag{2.34}$$

Si $\mid C_i \mid = 1$ entonces $s(i) = 0$.

CAPÍTULO 3

Diseño de la arquitectura para análisis de imágenes

En este capítulo se describe detalladamente el diseño realizado junto con la implementación correspondiente para conseguir los objetivos presentados en el primer capítulo. Además se presentan los resultados obtenidos durante el proceso de análisis y las diferentes configuraciones que se han usado para su obtención.

3.1. Introducción y manejo de la información

Como se ha descrito en capítulos anteriores, el objetivo del trabajo es realizar un análisis comparativo entre los algoritmos de aprendizaje y clasificación sobre texturas en imágenes aéreas. La implementación de los algoritmos y los diferentes análisis se han llevado a cabo en el entorno MatLab (2019), en concreto en la versión R2016B para Macintosh. En el apéndice B se describen brevemente los programas utilizados.

La clasificación de las diferentes texturas se realiza a nivel de píxel, por lo que lo primero es definir una estructura que albergue los diferentes píxeles de tal forma que constituyan las muestras a utilizar por los diferentes algoritmos de aprendizaje. En la figura 3.1 se muestra la disposición de los píxeles de una imagen en una lista para posteriormente pasar a la fase de entrenamiento o clasificación. Para una imagen de tamaño $w \times h$ la lista posee una longitud de $n = w * h$ píxeles. Así pues, en total se dispone de un número n de muestras p_i donde $i = 1...n$.

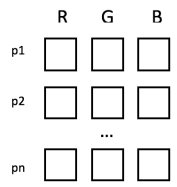


Figura 3.1: Lista de píxeles

En este caso, las imágenes aéreas a tratar son imágenes RGB. Por lo que cada píxel consta de tres valores: R (rojo), G(verde) y B(azul). Por tanto éstas son las componentes de los patrones que definen las muestras de entrada para el aprendizaje.

Para facilitar el análisis y poder realizar múltiples ejecuciones con diferentes parámetros, se ha desarrollado un programa principal, totalmente parametrizado, que permite realizar las fases de aprendizaje y clasificación de forma secuencial, además de permitir realizar una fase u otra indistintamente. Adicionalmente, permite escoger los algoritmos a utilizar para el entrenamiento y la clasificación.

Una vez realizado el aprendizaje, se utiliza una base de conocimiento compuesta por varias estructuras lo que permite almacenar los valores de las diferentes muestras ordenados por clases y los centros de cada clase para cada algoritmo, así como los parámetros involucrados en los diferentes algoritmos. Esta estructura se almacena en un fichero, lo que permite poder ser cargada y no perder la información de entrenamiento de una ejecución a otra.

Por otro lado, se han implementado los indicadores relativos a determinar la calidad de la partición *(Cluster Partition Index)*, descritos en la sección 2.3.6 que permitirán analizar las diferentes particiones y centros de cada clase para comparar los diferentes algoritmos entre sí.

3.2. Diseño

La propuesta para realizar la comparación de los distintos algoritmos de aprendizaje y clasificación de texturas en imágenes aéreas está compuesta de las siguientes etapas:

1. Aprendizaje
   - *a*) Particionado de datos
     - Algoritmo Fuzzy K-Means: número de clases conocido
     - Algoritmo Cuantización Vectorial (CV): umbral de partición
   - *b*) Cálculo de centros
     - Algoritmo Fuzzy K-Means
     - Algoritmo Cuantización Vectorial (CV)
     - Algoritmo Paramétrico de Bayes
     - Algoritmo Self-Organizing Map (SOM)
     - Algoritmo competitivo de Lloyd
2. Clasificación
   - Centros Fuzzy
   - Centros CV
   - Centros de Bayes
   - Centros Self-Organizing Map (SOM)
   - Centros Lloyd

Durante la etapa de aprendizaje y distribución de las muestras en clases mediante el algoritmo Fuzzy K-Means, una vez realizadas las particiones, se procede a calcular los centros de las mismas mediante el algoritmo Fuzzy K-Means, la media y matriz de covarianza de Bayes y los centros de las clases SOM.

En el caso de la partición mediante el algoritmo CV, tras tener las particiones identificadas y sus correspondientes muestras, se calculan los centros mediante el algoritmo CV y SOM y la media y matriz de covarianza de Bayes.

Una vez finalizado el proceso de aprendizaje, se dispone de las particiones con sus muestras así como todos los centros de las clases estimados con los diferentes algoritmos, por tanto se está en posición de pasar a la etapa de clasificación.

La figura 3.2 muestra el diagrama del diseño implementado para la realización del análisis y la comparación de algoritmos.

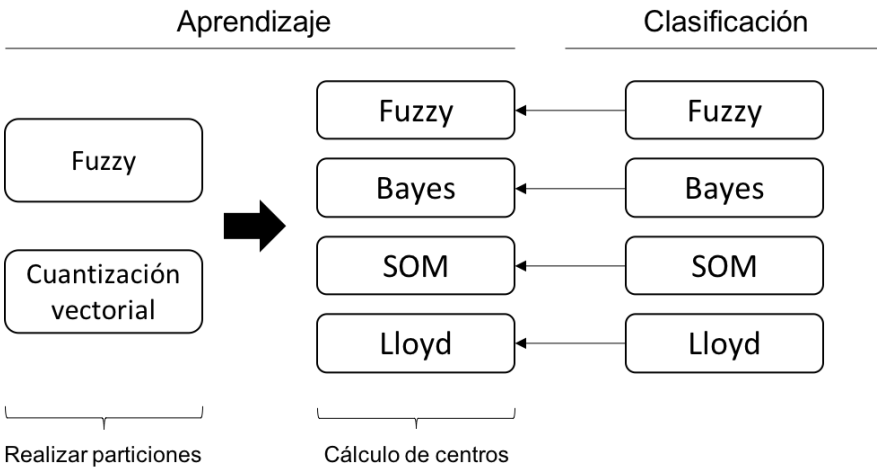


Figura 3.2: Escenario implementado

Según la metodología propuesta, para poder realizar el análisis, se ha construido un programa que permite realizar el aprendizaje y la clasificación proporcionando la información requerida por los algoritmos, donde se incluyen los diversos parámetros y se almacenan los resultados en una base de conocimiento. Veamos en detalle cada una de las fases.

El primer paso consiste en llevar a cabo la fase de aprendizaje y el punto de partida es realizar las particiones de los datos basándose en dos algoritmos de *clustering*: Fuzzy K-Means y CV. En el caso de Fuzzy K-Means se ha de indicar como parámetros el total de clases en las que dividir las muestras de entrenamiento, el peso y el umbral como criterio de terminación. Por otro lado, para realizar las particiones mediante el algoritmo CV, únicamente se ha de indicar el umbral T para separar los *clusters*, por lo que a diferencia de Fuzzy K-Means, no se establece el número de *clusters* y es el

algoritmo el que, en base al umbral establecido, crea un número determinado de *clusters* dependiendo de las muestras. En este proceso de partición también se calculan los centros de las clases para Fuzzy K-Means y CV. A continuación se procede al cálculo de los centros mediante los algoritmos de LLoyd, SOM y Bayes, estableciendo para cada uno de ellos los parámetros requeridos. En el caso de Lloyd se ha de indicar la razón de aprendizaje, el total de iteraciones y la tolerancia que determina la diferencia máxima, en términos de distancia, admisible para considerar que el algoritmo converge y por tanto los centros ya no se actualizan en sucesivas iteraciones. En el caso de SOM, se han de establecer los valores de la función que define el ancho de la función o zona de influencia, para ello se utiliza el valor alfa inicial y alfa final, además del total de iteraciones, el umbral y la tolerancia.

Una vez obtenidas las particiones y calculados los centros con los diferentes algoritmos, se pasa a la fase de clasificación. Para la clasificación el proceso es similar al de aprendizaje, en el caso del algoritmo de Fuzzy K-Means se ha de indicar el peso que se utiliza para calcular la distancia a los centros de los *clusters*. Para el resto de algoritmos no es requisito establecer ningún parámetro adicional, simplemente se calculan las distancias de los píxeles a los diferentes centros para encontrar la distancia mínima y asignar la clase correspondiente.

3.3. Análisis y estudio

El análisis a realizar se divide en dos fases: fase de aprendizaje y fase de clasificación. En primer lugar se aborda la fase de aprendizaje, donde se analiza y justifican los parámetros utilizados para cada algoritmo para después pasar a la fase de clasificación donde se comprueban los resultados de los algoritmos y el aprendizaje adquirido por los mismos.

3.3.1. Fase de aprendizaje

La fase de aprendizaje a su vez está divida en dos etapas: partición de los datos en *clusters* y cálculo de los centros mediante los diferentes algoritmos.

3.3.1.1. Partición de datos

Con el fin de determinar el número de clases óptimo y el umbral a utilizar en el algoritmo de CV, el siguiente análisis se basa en cómo de cohesionados están todos los elementos que pertenecen a una misma clase, concretamente mediante la métrica de suma de cuadrados intraclaster (SSW).

Se comienza con el algoritmo de Fuzzy K-Means para determinar el número de clases óptimo. Para ello, se realizan varios procesos de aprendizaje iterativos variando el número total de clases en las que se van a distribuir los datos o muestras disponibles.

En la figura 3.3 se observa cómo decrece la distancia entre elementos del mismo *cluster* conforme incrementa el número de clases. Si se dispusiera de tantas clases como elementos, la distancia entre los elementos de una misma clase sería 0, de ahí que a mayor número de clases mayor cohesión, si bien se ha de llegar a un compromiso entre el número de clases final y cómo de buenas son las particiones. Se puede observar cómo aparece un decremento del 25 % cuando se pasa de 2 a 3 clases y un decremento del 13 % cuando se pasa de 3 a 4 clases. A partir de 5 clases se observa que la distancia va disminuyendo pero de forma más estable, por tanto se puede decir que una división entre cuatro o cinco clases sería la más adecuada.

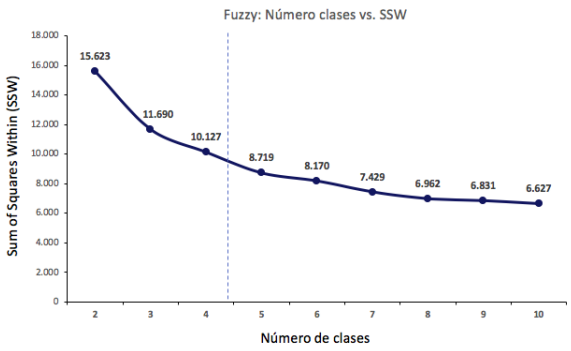


Figura 3.3: Evolución de SSW con el aumento de clases en Fuzzy.

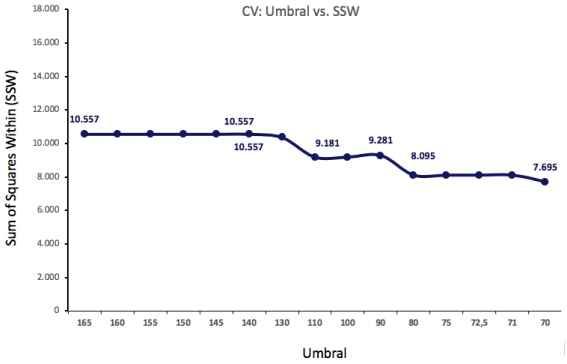


Figura 3.4: Evolución de SSW con el umbral en CV.

En cuanto al algoritmo CV, en la figura 3.4 se observa cómo se comporta el SSW conforme va disminuyendo el umbral, es decir, a menor umbral más clases se generan y menor es la distancia entre los elementos de una misma clase. Para un umbral de 110 se tiene un descenso del 11 % de distancia, por lo que para ese umbral los elementos de las clases creadas por CV están bien cohesionados. Además, para un umbral de 80 se aprecia otro descenso considerable, por lo que puede ser otra buena opción para realizar la partición. En el proceso de aprendizaje aplicado a imágenes aéreas estos umbrales corresponden a cuatro y seis clases respectivamente. En la figura 3.5 se presenta la correspondencia entre cada umbral y el total de clases que genera el algoritmo CV para dicho umbral.

Umbral	**165**	**160**	**155**	**150**	**145**	**140**	**130**	**110**	**100**	**90**	**80**	**75**	**71**	**70**
Clases	3	3	3	3	3	3	4	4	4	5	6	6	6	6

Figura 3.5: Correspondencia umbral y total de clases creadas por CV.

Por tanto, en base al análisis llevado a cabo, se puede determinar que cuatro es el número óptimo de clases para poder realizar el estudio para la evaluación de los diferentes algoritmos.

Una vez conocido el total de clases en las que se dividen los datos, se realiza un análisis basado en el coeficiente de Silhouette para determinar la consistencia de los

datos de cada *cluster*.

Como se observa en la figura 3.6 (a) para el algoritmo de Fuzzy el promedio de los coeficientes de Silhouette es similar en las cuatro clases y con un valor medio aproximado de 0,66 (el rango de valores que puede tomar el coeficiente de Silhouette se sitúa en el intervalo [-1,1], indicando -1 poca consistencia y 1 máxima consistencia). En cuanto a las particiones realizadas con CV, se observa como el coeficiente también es bastante similar a Fuzzy K-Means con un valor medio de 0,67.

En la figura 3.6 (b) se muestran los coeficientes de Silhouette pero con seis grupos en vez de cuatro. Como se puede observar, en la mayoría de los *clusters*, baja el coeficiente teniendo un total de $0,64$ para ambos algoritmos, lo que es inferior a la particiones con cuatro grupos. Por tanto cuatro es el número óptimo de clases para el conjunto de datos disponibles.

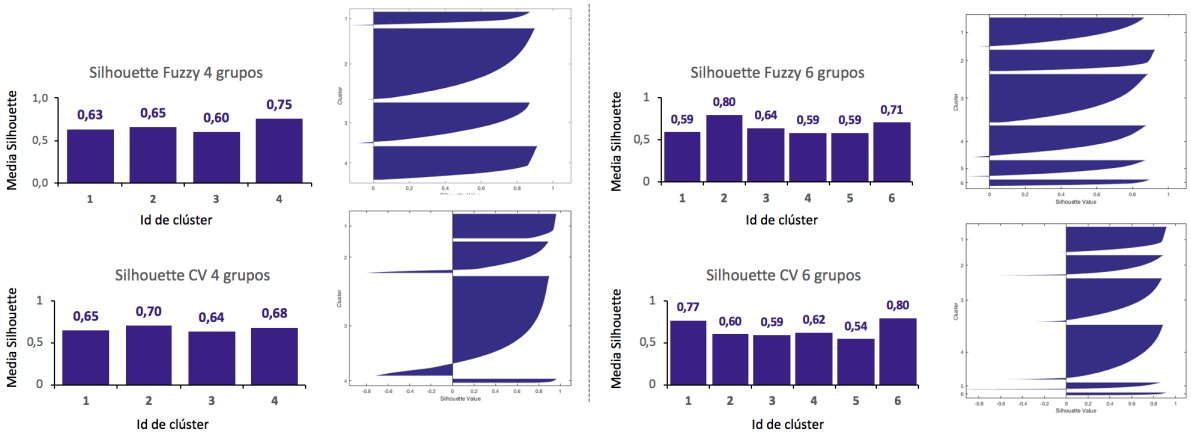


Figura 3.6: Coeficiente de Silhouette para las clases de Fuzzy y CV. (a) Con cuatro grupos; (b) Con seis grupos.

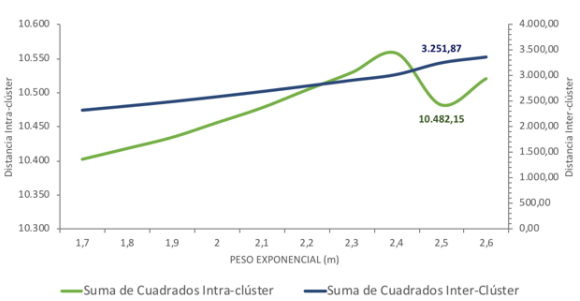
3.3.1.2. Cálculo de centros

Una vez obtenidas las particiones mediante ambos algoritmos, se pasa a la segunda fase de la etapa de aprendizaje. En esta etapa se lleva a cabo el cálculo de los centros de los *clusters*. En el caso del algoritmo Fuzzy K-Means y de CV los centros ya han sido calculados durante el proceso de partición, si bien se detalla con mayor nivel de detalle, a la vez que se explica su cómputo para el resto de algoritmos.

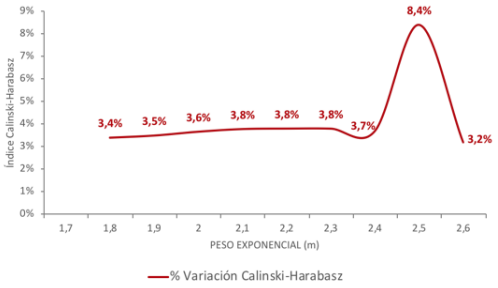
Fuzzy K-Means

Como se expone en la sección 2.3.1 para este trabajo se utiliza una variación del algoritmo Fuzzy K-Means en la que los centros iniciales utilizados para realizar las particiones son asignados de forma pseudo-aleatoria basada en Balasko y col (2006). En Fuzzy K-Means el parámetro *peso exponencial (m)* tiene un papel fundamental para el cálculo de los centros durante el proceso de partición de datos. Con la finalidad de

establecer el valor óptimo de m para obtener los centros de los *clusters* se han calculado las distancias intra-*cluster (Sum of Square Within)*, las distancias inter-*cluster (Sum of Square Betweeen)* y el índice de Calinski-Harabasz para diferentes valores de m. La figura 3.7 muestra el comportamiento de los índices SSW, SSB y CH con el incremento del peso exponencial en Fuzzy K-Means.



(a) SSB y SSW vs. Peso Exponencial



(b) Variación del índice de Calinski-harabasz vs. Peso Exponencial

Figura 3.7: Comportamiento de los índices SSW, SSB y CH con el incremento del peso exponencial en Fuzzy K-Means

Como se observa en la figura 3.7a conforme se incrementa el valor del peso exponencial aumenta la distancia entre *clusters* (SSB), del mismo modo incrementa la distancia de los elementos de un mismo *cluster* (SSW). Para el valor de $m = 2{,}5$ se observa una disminución de la distancia intra-*cluster* por lo que puede ser un valor adecuado. Con el fin de determinar la relación entre ambas distancias en una única variable se ha calculado la evolución del índice de Calinski-Harabasz (CH) con el incremento del peso exponencial. Se puede observar cómo para $m = 2,5$ el índice de CH tiene un incremento del $8,4\,\%$ frente al $3{,}6\,\%$ de media para el resto de valores de m. Por tanto, se ha comprobado que un valor de 2,5 como peso exponencial m para Fuzzy K-means es adecuado para el cálculo de los centros.

Bayes

Como se detalla en la sección 2.3 el clasificador Bayesiano se basa en dos parámetros: la media y matriz de covariancia de las particiones. En el proceso de clasificación se utilizan ambos en la función de densidad de probabilidad que define cada una de las clases, para calcular las distancias de un elemento a todos los *clusters* y determinar el más cercano para su clasificación.

Dada la naturaleza del algoritmo, no es necesario establecer ningún parámetro previo, simplemente se ha de explorar cada partición para calcular la media y covarianzas de todos los elementos que las componen. En el caso de Bayes se calculan ocho centros en total, uno por cada partición de Fuzzy K-Means y otro por cada partición de CV.

Cuantización Vectorial

En el caso de CV el cálculo de centros se lleva a cabo durante la fase de partición. Cuando un elemento se añade a una clase existente se calcula el centro del agrupamiento mediante la media aritmética de todos los elementos que componen dicho agrupamiento.

Algoritmo competitivo de Lloyd

Para el cálculo de centros mediante Lloyd inicialmente se asigna la media de cada partición como centro de dicha partición. A continuación comienza la competición entre centros en base a la razón de aprendizaje. Para este estudio se ha utilizado una única razón de aprendizaje para todos los *clusters*. Con la finalidad de obtener los centros que mejor representan a cada grupo, es importante establecer una razón de aprendizaje adecuada. Para la elección de la razón de aprendizaje se realiza una comparación del valor del indice de Calenski-Harabasz (CH) tanto para los agrupamientos de Fuzzy K-Means como de CV. En la figura 3.8 se muestra el comportamiento del índice Calinski-Harabasz con el incremento de la razón de aprendizaje.

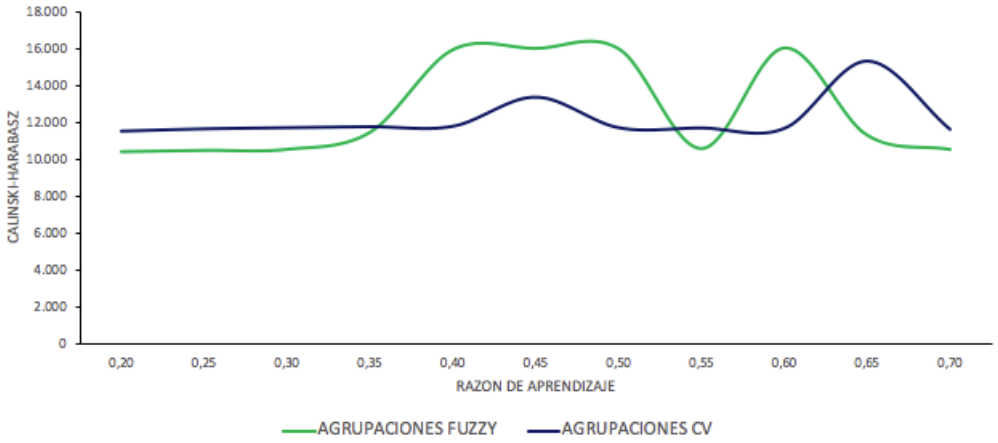


Figura 3.8: Evolución indice Calinski-Harabasz para diferentes razones de aprendizaje en el algoritmo de Lloyd.

Como se observa en la figura 3.8 a partir del valor 0,35 el índice de CH se incrementa teniendo un máximo relativo en 0,45 para las agrupaciones de CV y manteniéndose estable para las agrupaciones de Fuzzy K-Means. Por tanto, una razón de aprendizaje de 0,45 se considera un valor apropiado para el cálculo de los centros mediante el algoritmo competitivo de Lloyd.

Self-Organizing Map

El algoritmo SOM requiere de cinco parámetros de entrada: ancho inicial de la vecindad α_i, ancho final de la vecindad α_f, número total de iteraciones, umbral para

determinar si un patrón pertenece a una clase y tolerancia final. Para la elección de los valores de cada parámetro se realizan quince ejecuciones con diferentes valores para escoger los parámetros que mejor se ajusten a las imágenes a tratar en este estudio. Uno de los factores a tener en cuenta es el tiempo de convergencia del algoritmo, para ello se ha establecido un total de 1,000 iteraciones y una tolerancia de e^{-5}. Para el parámetro α_i se establece el valor e^{-2} y e^{-4} para α_f . En el caso del umbral y la tolerancia se usan los valores de e^{-4} y e^{-5} respectivamente.

3.3.2. Fase de clasificación

La fase de clasificación consiste en recorrer la imagen a clasificar píxel a píxel y, en base a los centros calculados en la fase de aprendizaje, calcular la distancia al centro de cada clase para escoger el más cercano, la clase de ese centro será la clase del pixel a clasificar. A continuación se detalla para cada algoritmo cómo se calculan dichas distancias y los parámetros necesarios para ello.

Fuzzy K-Means

Para la clasificación de un píxel mediante el algoritmo de Fuzzy K-Means se han de calcular los grados de pertenencia a cada uno de los centros de las diferentes particiones mediante la ecuación (2.4). Para realizar dicho cálculo es necesario establecer un valor para el *peso exponencial*, para ello se ha utilizado el mismo valor que en el proceso de aprendizaje.

Finalmente, una vez calculados los grados de pertenencia el píxel es asignado a aquella clase cuyo grado de pertenencia sea mayor.

Clasificador Bayesiano

En el caso particular del clasificador Bayesiano, se calcula la probabilidad de que un píxel pertenezca a cada una de las clases aplicando la ecuación (2.21) mediante las matrices de covarianza y medias de cada clase. Finalmente, para cada píxel se escoge la clase con mayor probabilidad.

CV, Lloyd y SOM

En el caso de los algoritmos CV, LLoyd y SOM para la clasificación de los píxeles se utiliza la distancia euclídea como métrica, de tal modo que un píxel pertenecerá a aquella clase donde la distancia entre el píxel y el centro de la clase sea mínima.

CAPÍTULO 4

Resultados

En este capítulo se muestran los resultados obtenidos basados en el diseño y análisis propuestos en el capítulo anterior. En primer lugar se describen las imágenes utilizadas para el análisis de los resultados. En segundo lugar se realiza una comparativa entre Fuzzy K-Means y CV para evaluar las particiones iniciales obtenidas mediante ambos algoritmos. Finalmente se comparan los resultados de todos los algoritmos en la fase de clasificación.

4.1. Descripción de las imágenes

Para el análisis de los resultados se han utilizado un total de 40 imágenes, que se han dividido en dos grupos para realizar los experimentos durante las correspondientes fases de entrenamiento y clasificación, cada uno de ellos formado por 20 imágenes. La ubicación de las imágenes en cada uno de estos dos grupos se ha realizado de forma aleatoria con el fin de evitar influencias que pudieran afectar a la fase de entrenamiento. La resolución de todas ellas es de 500x500x3, por consiguiente cada una posee 250.000 píxeles con sus correspondientes valores RGB.

Las imágenes fueron proporcionadas en el año 2007 por la empresa DIMAP (DIGITAL IMAGE PRO, S.L.) provenientes del Servicio Territorial de Galicia (SITGA).

Para la fase de aprendizaje se han seleccionado, también aleatoriamente, el 40 % de los píxeles de cada imagen lo que supone 105 píxeles por imagen, haciendo un total de 2x106 píxeles en total para el conjunto de las 20 imágenes disponibles para esta fase.

Cada una de las 20 imágenes pertenecientes al grupo de la fase de clasificación se ha procesado por separado. En este sentido se han identificado dos subgrupos con características específicas desde el punto de vista del análisis de resultados:

Grupo A: caracterizado por la diversidad de parcelas de terreno de tamaño lo suficientemente grande para poder comparar visualmente la clase en la que se clasifican

sus píxeles y su heterogeneidad.

Grupo B: caracterizado por la abundancia de zonas verdes delimitadas por zonas áridas con algunos caminos.

Los resultados que se muestran en la sección 4.3 se realizan con dos de las imágenes representativas de cada grupo teniendo en cuenta que los resultados para el resto de imágenes de cada subgrupo son similares, sin aportar aspectos o elementos significativos o relevantes que merezcan un análisis individualizado.

4.2. Comportamiento de los índices de validación en fase de aprendizaje

4.2.1. Evaluación de particiones

Como se describe en la sección 3.3.1.1 la partición de datos se realiza con dos algoritmos diferentes: Fuzzy K-Means y CV. En base a las ejecuciones realizadas con los parámetros definidos en dicha sección, a continuación se presenta una comparativa de diferentes métricas e índices para ambos algoritmos.

Como se puede observar en la tabla de la figura 4.1a el *Diámetro máximo* de las particiones obtenidas mediante Fuzzy K-Means es de $1,53$ frente a $1,57$ de CV. En la tabla de la figura 4.1b se observa que para las clases 2 y 4 las particiones obtenidas mediante Fuzzy K-Means el *diámetro máximo* con respecto a las particiones de CV es inferior, concretamente difieren en $0,16$ y $0,08$ respectivamente. Por el contrario, para las clases 1 y 3, CV tiene un *diámetro máximo* inferior con una diferencia de $0,05$ y $0,02$ por lo que no es muy representativa.

	Diámetro máximo	Separación Mínima
Fuzzy K-Means	1,53	0,0008
Cuantizacion Vectorial	1,57	0,0005

(a) *Diámetro máximo* y *Separación mínimina* Fuzzy K-Means y CV

Clase	*Fuzzy K-Means*	*Cuantizacion Vectorial*
1	1,53	1,48
2	1,41	1,57
3	1,44	1,42
4	1,47	1,55

(b) *Diámetro máximo* por clase

Figura 4.1: Índices de validación de *cluster* para los algoritmos Fuzzy K-Means y CV

Por otro lado, se observa que para la métrica de *separación mínima* las particiones de Fuzzy K-Means están más separadas que las obtenidas mediante CV, por lo que se puede concluir que las particiones obtenidas mediante Fuzzy K-Means son más apropiadas para el conjunto de muestras tratadas.

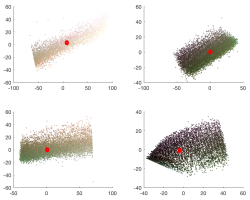
4.2.2. Evaluación de centros de *cluster*

Una vez analizadas las particiones de Fuzzy K-Means y CV se procede a analizar el resultado del cálculo de centros con los diferentes algoritmos, a saber: Fuzzy K-Means,

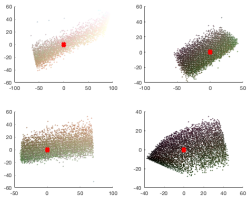
Bayes, SOM y LLoyd. En primer lugar se comparan los centros obtenidos sobre las particiones obtenidas mediante Fuzzy K-Means. A continuación se analiza el cálculo de centros sobre las particiones de CV.

Centros en particiones Fuzzy K-Means

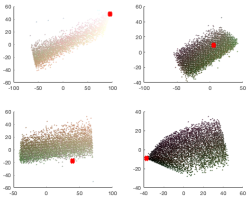
En primer lugar, se calculan los centros mediante Fazzy K-Means (vector de centros v), Bayes, SOM y Lloyd sobre las particiones obtenidas mediante Fuzzy K-Means. En la figura 4.2 se muestra la distribución de los patrones o muestras en cada partición Fuzzy K-Means y los centros obtenidos mediante los algoritmos que se indican.



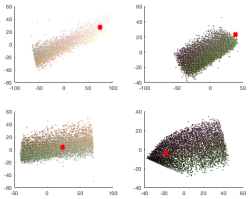
(a) Centros Fuzzy K-Means



(b) Centros Bayes



(c) Centros SOM



(d) Centros Lloyd

Figura 4.2: Centros en particiones Fuzzy K-Means

A simple vista se puede observar cómo los centros obtenidos mediante el algoritmo Fuzzy K-Means (a) y el algoritmo de Bayes (b) son muy similares y en su mayoría ubicados en una posición centrada respecto de la distribución de las muestras en las particiones. Por el contrario, en el caso de los algoritmos SOM (c) y Lloyd (d) los centros en su mayoría se localizan en alguno de los extremos de la partición.

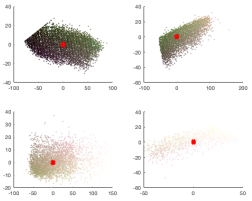
Atendiendo a los resultados obtenidos mediante las diferentes métricas de validación de *cluster* mostrados en la tabla de la figura 4.3 se observa que los centros obtenidos mediante el algoritmo de Bayes tienen una mayor cohesión, si bien es cierto no tienen la mejor separación, el índice de Ball-Hall y Calinski-Harabasz toman los mejores valores, y con diferencia, para Bayes frente al resto de algoritmos. Por tanto, en lo que respecta a particiones Fuzzy K-Means, se puede concluir que los centros obtenidos mediante el algoritmo de Bayes son los más adecuados para representar cada una de las particiones y ser utilizados en la fase de clasificación para el cálculo de distancias.

	SSW (Cohesión)	SSB (Separación)	Ball-Hall	Hartigan	Xu	Calinski-Harabasz
Fuzzy K-Means	4.953	5.323	1.238	0,03	-8,60	17.059
Bayes	755	4.958	189	0,82	-9,83	104.235
SOM	2.396	11.402	599	0,68	-9,08	75.548
Lloyd	2.346	8.687	586	0,57	-9,09	58.785

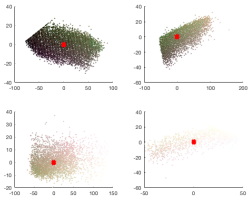
Figura 4.3: Índices de validación de *cluster* sobre particiones Fuzzy K-Means

Centros en particiones CV

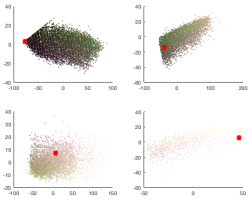
En la figura 4.4 se muestran los resultados obtenidos para los centros calculados sobre particiones CV de los algoritmos indicados.



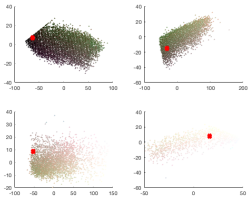
(a) Centros Fuzzy K-Means



(b) Centros Bayes



(c) Centros SOM



(d) Centros Lloyd

Figura 4.4: Centros en particiones CV

Del mismo modo que en las particiones Fuzzy K-Means, los centros obtenidos mediante CV y Bayes se localizan en el centro de las particiones, a diferencia de los obtenidos mediante SOM y CV, los cuales se sitúan, en su mayoría, hacia los extremos de la partición.

En la tabla de la figura 4.5 se observa que, para la mayoría de índices, el algoritmo de Bayes toma los mejores valores a excepción del índice SSB (Separación), el cual toma el mismo valor tanto para los centros obtenidos por CV y por Bayes.

	SSW (Cohesión)	SSB (Separación)	Ball-Hall	Hartigan	Xu	Calinski-Harabasz
CV	5.139	4.589	1.285	-0,05	-8,58	14.174
Bayes	1.125	4.589	281	0,61	-9,57	64.764
SOM	3.641	10.508	910	0,46	-8,81	45.812
Lloyd	5.550	10.102	1.387	0,26	-8,53	28.896

Figura 4.5: Índices de validación de *cluster* sobre particiones CV

Comparativa global

En la figura 4.6 se muestra una comparativa global de todos los índices para cada algoritmo sobre particiones Fuzzy K-Means y CV, con la finalidad de determinar qué algoritmos son los más adecuados para la obtención de las particiones y para el cálculo de los centros de las mismas.

	SSW (Cohesión)	SSB (Separación)	Ball-Hall	Hartigan	Xu	Calinski-Harabasz
Fuzzy K-Means	4.953	5.323	1.238	0,03	-8,60	17.059
CV	5.139	4.589	1.285	-0,05	-8,58	14.174
Bayes Fuzzy K-Means	755	4.958	189	0,82	-9,83	104.235
Bayes CV	1.125	4.589	281	0,61	-9,57	64.764
SOM Fuzzy K-Means	2.396	11.402	599	0,68	-9,08	75.548
SOM CV	3.641	10.508	910	0,46	-8,81	45.812
Lloyd Fuzzy K-Means	2.346	8.687	586	0,57	-9,09	58.785
Lloyd CV	5.550	10.102	1.387	0,26	-8,53	28.896

Figura 4.6: Índices de validación de *cluster* sobre particiones CV y Fuzzy K-Means

Como se puede observar para las particiones obtenidas mediante Fuzzy K-Means y centros calculados con el algoritmo de Bayes, la mayoría de los índices poseen el mejor valor. Por el contrario, para las particiones obtenidas mediante CV y centros calculados con el algoritmo competitivo de Lloyd, la mitad de los índices contienen el peor valor. Se puede concluir que las particiones obtenidas mediante el algoritmo Fuzzy K-Means y cuyos centros son calculados mediante Bayes, son las más adecuadas para el tipo de muestras utilizadas en este estudio.

4.3. Fase de clasificación

Para la clasificación de un píxel se utilizan los centros de cada una de las particiones para calcular la distancia y determinar a qué clase pertenece dicho píxel. En el caso de Fuzzy K-Means, se calculan los valores de pertenencia en base al peso exponencial, para el resto de algoritmos se hace uso de la distancia Euclídea.

A continuación se analiza el proceso de clasificación para varias imágenes, mostrando los resultados obtenidos, medidas de error e identificación de texturas.

4.3.1. Imagen del subgrupo A

A continuación se describen los resultados obtenidos con la imagen representativa del subgrupo A.

En la figura 4.7 se muestra la imagen original a partir de la cual se procede a la extracción de texturas y clasificación en cuatro clases diferentes.



Figura 4.7: Imagen subgrupo A

En la figura 4.8 se observa el resultado obtenido para cada uno de los algoritmos aplicados. La primera fila contiene las texturas extraídas basándose en las particiones Fuzzy K-Means y aplicando los diferentes algoritmos para el cálculo de los centros. Por el contrario, la segunda fila contiene las texturas basándose en particiones calculadas mediante el algoritmo de CV.

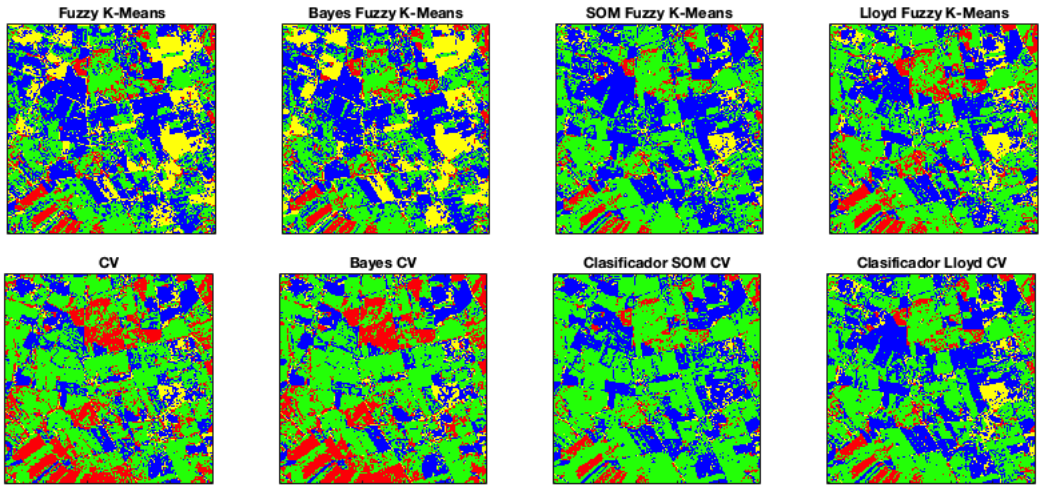


Figura 4.8: Texturas de la imagen del subgrupo A clasificadas mediante diferentes algoritmos

Visualmente cabe destacar que para los algoritmos de Fuzzy K-Means y Bayes Fuzzy K-Means las texturas de la clase 4 (véase figura 4.9) son más abundantes en comparación con el resto de algoritmos. Esta diferencia es debida a los centros calculados en el

proceso de aprendizaje, ya que para ambos algoritmos en las tres primeras clases predomina el color verde, quedando el color marrón (terrenos más áridos) para la cuarta clase. Por el contrario, en el resto de algoritmos en su mayoría hay dos clases donde predomina el color verde y dos los colores marrones.

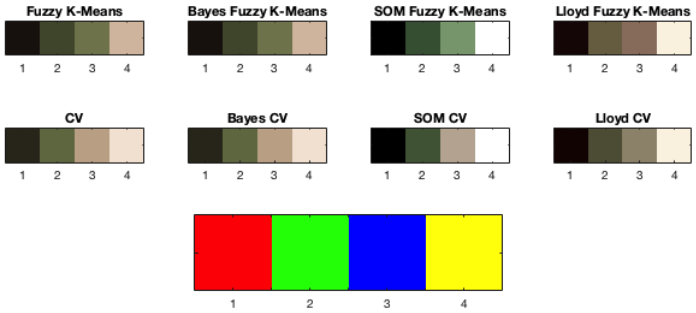


Figura 4.9: Correspondencia colores clases y centros de particiones

Con la finalidad de evaluar cada una de las clasificaciones, se calculan las distancias de cada píxel de la imagen original con el centro de la clase en la que ha sido clasificado. En la figura 4.10 se puede comprobar cómo para el algoritmo de Bayes Fuzzy K-Means el promedio de las todas las distancias toma el valor mínimo frente al resto de algoritmos.

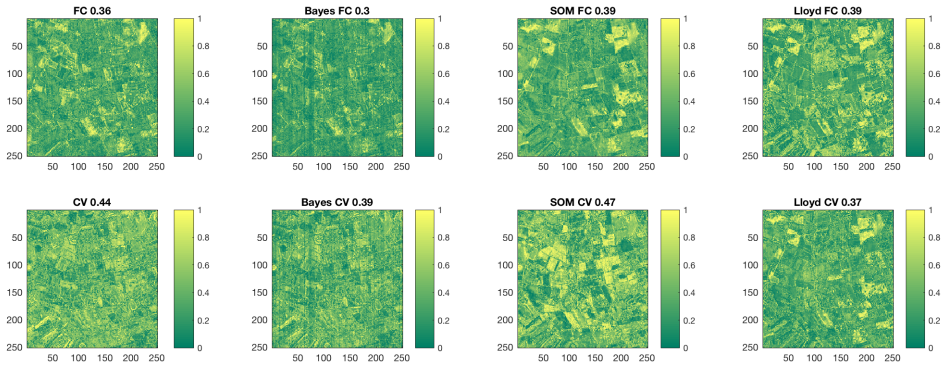


Figura 4.10: Distancia entre píxel original y centro de la clase donde ha sido clasificado.

Finalmente, si se define un umbral por el que se considera que un píxel no ha sido clasificado correctamente, se puede comprobar cuáles son las texturas donde se está cometiendo mayor error. En la figura 4.11 se muestran aquellas texturas con un umbral superior al 0,7.

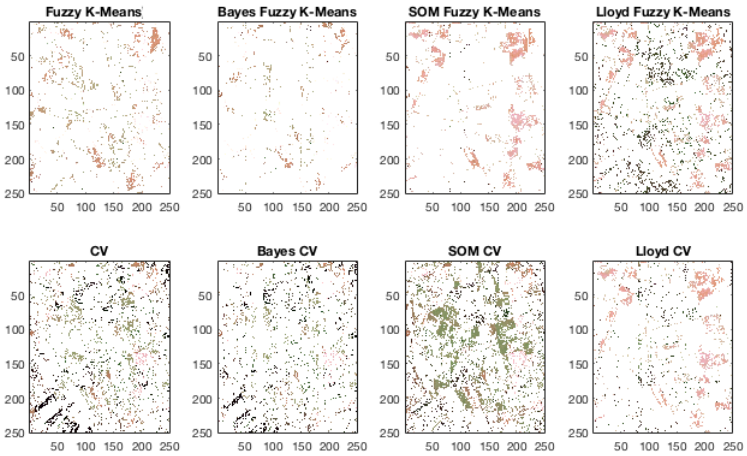


Figura 4.11: Píxeles cuya distancia supera el umbral definido

En el apéndice A.1 se muestran todas las texturas clasificadas con cada algoritmo para esta imagen.

4.3.2. Imagen del subgrupo B

A continuación se describen los resultados obtenidos con la imagen representativa del subgrupo B.

En la figura 4.12 se muestra la imagen original a partir de la cual se procede a la extracción de texturas y clasificación en cuatro clases diferentes.



Figura 4.12: Imagen subgrupo B

En la figura 4.13 se muestran los resultados obtenidos para cada uno de los algoritmos aplicados. Al igual que en análisis de la imagen de la sección anterior, la primera fila contiene las texturas extraídas basándose en las particiones Fuzzy K-Means y aplicando los diferentes algoritmos para el cálculo de los centros. Por el contrario, la segunda fila contiene las texturas basándose en particiones calculadas mediante el algoritmo de CV.

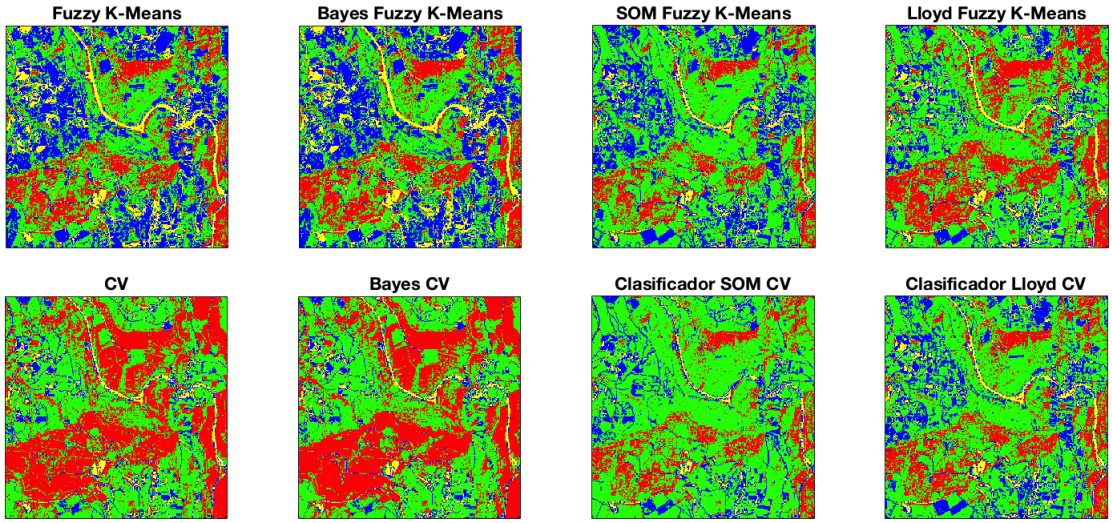


Figura 4.13: Texturas de la imagen del subgrupo B clasificadas mediante diferentes algoritmos

En base a las imágenes de la figura 4.13 a nivel visual se observa cómo los algoritmos con particiones Fuzzy K-Means presentan mejores resultados que los basados en CV. Dentro los algoritmos sobre particiones Fuzzy K-Means, se observa cómo Fuzzy K-Means y Bayes Fuzzy K-Means son los dos que mejores resultados ofrecen. Analíticamente se puede comprobar dicha afirmación calculando la distancia media de cada píxel al centro de la clase que se le ha asignado. En la figura 4.14 se muestran las distancias de todos los píxeles al centro de su clase y la distancia media total de la imagen. Como se puede comprobar, para Bayes con Fuzzy K-Means la distancia media es de $0,27$, siendo ésta la menor de todas las distancias en comparación con los demás algoritmos.

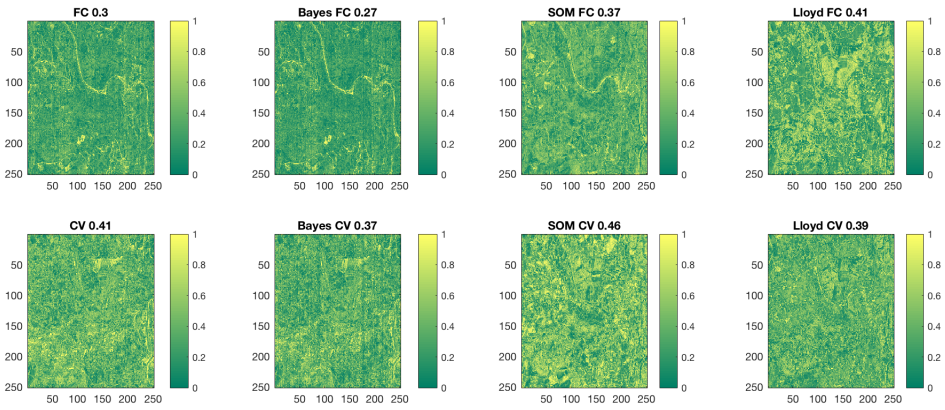


Figura 4.14: Distancia entre píxel original y centro de la clase donde ha sido clasificado.

Estableciendo un umbral máximo de $0,7$, en la figura 4.15 se muestran aquellos píxeles que superan dicho umbral y por tanto en los que más error se está cometiendo en su clasificación.

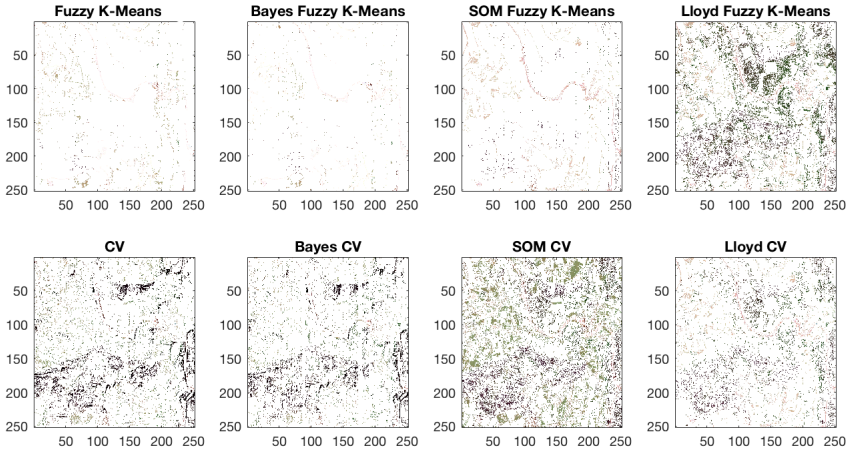


Figura 4.15: Píxeles cuya distancia supera el umbral definido

Como se puede observar, para el clasificador paramétrico de Bayes sobre particiones Fuzzy K-Means, el número de píxeles que superan el umbral definido es muy inferior al del resto de algoritmos. En el apéndice A.2 se muestran todas las texturas clasificadas con cada algoritmo sobre esta imagen.

CAPÍTULO 5

Conclusiones y trabajo futuro

En este trabajo se han aplicado cinco algoritmos de clasificación para identificar texturas naturales en imágenes aéreas mediante texturas que poseen información espectral, en este caso el color en el formato RGB. No obstante, conviene reseñar en este sentido que no se ha considerado información espacial de carácter contextual. En primer lugar se obtienen las particiones de datos mediante los algoritmos Fuzzy K-Means y Cuantización Vectorial. A continuación, se calculan los centros de dichas particiones mediante el algoritmo paramétrico de Bayes, Self-Organizing Map y Lloyd. Finalmente, mediante el uso de índices de validación de *cluster*, que permiten medir la calidad de las particiones obtenidas, se compara el desempeño de los algoritmos entre sí. En base a dicho análisis, se extraen las siguientes conclusiones:

- Para los ejemplos de imágenes y texturas analizados se ha determinado que la mejor partición se obtiene con cuatro clases.
- Para la obtención de las particiones iniciales en el proceso de aprendizaje, el Fuzzy K-Means presenta mejores resultados que el proceso Cuantización Vectorial.
- Los algoritmos basados en redes neuronales (SOM y Lloyd) no son muy apropiados para la clasificación de texturas en imágenes aéreas.
- El algoritmo paramétrico de Bayes sobre particiones obtenidas mediante Fuzzy K-Means es el que mejor resultado ofrece en comparación con los demás algoritmos.

Vistos los resultados obtenidos, y el hecho de que existen otras posibilidades de estudio, se plantean los siguientes puntos como trabajo futuro:

- Se propone la realización de una comparativa de los diferentes algoritmos sobre imágenes representadas en otros modelos de color como HSV o CieLab, por poner dos ejemplos significativos.
- Se plantea el análisis del comportamiento de otros índices de validación diferentes a los abordados en este estudio.

- Se propone la creación de una base de datos de imágenes ground-truth para determinar el comportamiento de los métodos frente a criterios de expertos, lo que permitiría precisar más los resultados de clasificación a nivel de píxel mediante la utilización de matrices de confusión o curvas ROC con expresión de positivos y negativos verdaderos (TP/TN, true positives/negatives) y falsos (false positives/negatives).
- Se propone el estudio de más métodos de clasificación, incluyendo las redes neuronales del tipo retro-propagación (Pajare y Cruz, 2007) o incluso la combinación de clasificadores (Pajares y col., 2010).
- Se plantea también el uso de redes neuronales convolucionales dentro del paradigma de aprendizaje profundo, para segmentación de imágenes, pero ya desde un punto de vista conceptual a nivel de clasificación frente a los algoritmos utilizados. En este caso, la segmentación semántica se perfila como la opción candidata a utilizar (Marmanis y col., 2016).
- Como se ha indicado previamente, el trabajo se ha centrado exclusivamente en análisis de color, por lo que como acción de futuro se propone la incorporación de análisis contextual de carácter espacial donde la información de color se puede complementar mediante el uso de descriptores tales como las matrices de coocurrencia entre otras (Pajares y Cruz, 2007).

Bibliografía

[1] Alegre, E., Pajares, G., Escalera, A. (Eds.) (2016). *Conceptos y Métodos en Visión por Computador.* Comité Español de Automática.
Disponible on-line: http://intranet.ceautomatica.es/sites/default/files/upload/8/files /ConceptosyMetodosenVxC.pdf (accedido Marzo 2019).

[2] Arbelaitz, O., Gurrutxaga I., Muguerza, J., Pérez, J.M., Perona, I (2013). *An extensive comparative study of cluster validity indices.* Pattern Recognition. 46(1), 243–256.

Disponible on-line: https://ccc.inaoep.mx/ ariel/2013/An %20extensive %20comparative %20study %20of %20cluster %20validity %20indices.pdf (accedido Enero 2019)

[3] Balasko, B., Abonyi, J. and Feil, B. (2006). *Fuzzy clustering and data analysis toolbox for use with Matlab, Dpt. of Process Engineering.* University of Veszprem, Veszprem, Hungary.
Disponible on-line: ftp://ftp.unicauca.edu.co/cuentas/fiet/docs/DEIC/Materias /computacion %20inteligente/parte %20II/semana12/clustering/mfiles /Clustering-Toolbox/FuzzyClusteringToolbox.pdf (accedido Abril 2019)

[4] Bensaid, A.M., Hall, L.O., Bezdek, J.C., Clarke, L.P., Silbiger, M.L., Arrington, J.A. and Murtagh, R.F. (1996). *Validity-guided (Re) Clustering with applications to image segmentation* IEEE Trans. Fuzzy Systems, 4, 112-123.

[5] Bezdek, J.C., Pal, N.R. (1998). *Some new indexes of cluster validity.* IEEE Trans. Syst. Man Cybern. Part B Cybern. 28(3), 301–315.

[6] Bianco, S., Cusano, C., Napoletano, P., Schettini (2017). *Improving CNN-Based Texture Classification by Color Balancing* Journal of Imaging, 15.

[7] Bishop, C.M. (1995). *Neural Network for Pattern Recognition.* Clarendon Press, New York.

[8] Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., Dougherty, E.R. (2007) *Model-based evaluation of clustering validation measures. Pattern Recognition.* 40(3), 807–824.

[9] Calinski, T., Harabasz, J.. *A dendrite method for cluster analysis, Communications in Statistics.* 3:1, 1-27, 1974.

[10] Charrad, M., Ghazzali, N., Boiteau, V., Niknafs, A. (2014). *An R Package for Determining the Relevant Number of Clusters in a Data Set.* Journal of Statistical Software, 61(6).

[11] Cherkassky, V. and Mulier, F. (1998). *Learning from Data: Concepts, Theory and Methods* Wiley, New-York.

[12] Davies, D.L., Bouldin, D.W. (1979). *A Cluster Separation Measure.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1(2), 224-227.

[13] Desgraupes, B. (2017). *Clustering Indices* Cran R-Project.
Disponible on-line: https://cran.r-project.org/web/packages/clusterCrit /vignettes/clusterCrit.pdf (accedido Marzo 2019)

[14] Duda, R.O., Hart, P.E., Stork, D.G. (2012). *Pattern Classification.* Wiley, Hoboken.

[15] Dunn, J.C. (1974). *Well Separated Clusters and Optimal Fuzzy Partitions.* Journal Cybernetics, 4(1), 95-104.

[16] Dunn, J.C. (1976). *Indices of partition fuzziness and the detection of clusters in large data sets.*

[17] Dunn, J.C. (1976). *In Fuzzy Automata and decision process.* M.M. Gupta, Ed. New York, Elsevier.

[18] Escobedo, M.T., Salas, J.A (2008). *P. CH. Mahalanobis y las aplicaciones de su distancia estadística.* CULCyT, 27.

[19] Hartigan J. A., Wong M. A. (1979). *Algorithm AS 136: A K-Means Clustering Algorithm.* Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1).

[20] Huang, H.J., Hsu, C.N. (2002). *Bayesian classification for data from the same unknown.* IEEE Transactions on Systems, Man, and Cybernetics, Part B, 32(2), 137-145.

[21] Kim, D.W., Lee, K.H., Lee, D. (2003). *Fuzzy cluster validation index based on inter-cluster proximity.* Pattern Recognition Letters, 24, 2561-2574.

[22] Kohonen, T. (1984). *Self-Organization and Associative Memory.* Springer-Verlag, Berlin.

[23] Kohonen, T. (1990). *Self-Organization Map.* Proc. of the IEEE, 78(9), 1464-1480.

[24] Legány, C., Juhász, S., Babos, A. (2006) . *Cluster Validity Measurement Techniques* Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, 15-17 (388-393).
Disponible on-line: https://pdfs.semanticscholar.org/ 581c71da74bd3baa06693cc6d0751e7c60f81bb3.pdf (accedido Febrero 2019)

[25] Linde, Y., Buzo, A. and Gray, R.M. (1980). *An Algorithm for Vector Quantization Design.* IEEE Trans. Communications, 28(1), 84-95.

[26] Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J. (2010). *Understanding of Internal Clustering Validation Measures* IEEE International Conference on Data Mining. Disponible on-line (Mar-19): http://datamining.rutgers.edu/publication /internalmeasures.pdf (accedido Abril 2019)

[27] Lloyd, S.P. (1957). *Least squares quantization in PCM's.* Bell Telephone Laboratories Paper, Murray Hill, NJ.

[28] Marmanis, D., Wegner, J.D., Galliani, S., Schindler, K., Datcu, M., Stilla, U. (2016). *Semantic segmentation of aerial images with an ensemble of CNNs.* ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci., 3, 473.

[29] Max, J. (1970). *Quantizing for minimum distortion.* IRE Trans. Info. Theory, 6, 7-12.

[30] Pajares, G., Cruz, J. (2007). *Visión por Computador: Imágenes digitales y Aplicaciones.* Segunda Edición. RA-MA, Madrid.

[31] Pajares, G., Guijarro, M., Ribeiro, A. (2010). *A Hopfield Neural Network for combining classifiers applied to textured images.* Neural Networks 23 (2010) 144-153.

[32] Pajares, G., Cruz, J.M. (2011). *Aprendizaje Automático: un enfoque práctico.* RA-MA, Madrid.

[33] Pajares, G., Cruz, J. (2007). *Visión por Computador: Imágenes digitales y Aplicaciones.* RA-MA, Madrid.

[34] Patterson, D.W. (1996). *Artificial Neural Networks.* Singapore, Prentice-Hall, Englewood Cliffs.

[35] Qinpei, Z., Xu, M., Fränti, P. (2009). *Sum-of-Squares Based Cluster Validity Index and Significance Analysis.* ICANNGA.

[36] Rojas, J.C., Mora, M., Santos, M. (2014). *New Version of Davies-Bouldin index for clustering validation based on hyper rectangles.* 6th Chilean Conference on Pattern Recognition (CCPR).

[37] Rousseeuw, P. (1987). *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis* J. Comput. Appl. Math, 20(1), 53–65.

[38] Rui, A., Sousa, J.M.C. (2006) . *Comparison of fuzzy clustering algorithms for classification* Evolving Fuzzy Systems, 112-117.

[39] Slonimn, N., Aharoni, E., Crammer, K. . *Hartigan's K-Means Versus Lloyd's K-Means – Is It Time for a Change?.* Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence.
Disponible on-line: https://pdfs.semanticscholar.org/d474/7de9e39a5e5b1d509 bf9b58d19b36df9c416.pdf (accedido Enero 2019)

[40] Villagra, A., Guzmán A., Pandolfi D., Leguizamón G.. *Análisis de medidas no-supervisadas de calidad en clusters obtenidos por K-means y Particle Swarm Optimization.* Universidad Nacional de la Patagonia Austral, Unidad Academica Caleta Olivia
Disponible on-line: https://www.palermo.edu/ingenieria/Pdf2010/CyT9/10.pdf (accedido Enero 2019)

[41] Wu, J., Xiong, H., Chen, J. (2009). *Adapting the right measures for k-means clustering* KDD, 877–886.

[42] Xie X.L., Beni, G. (1991). *Validity measure for fuzzy clustering.* IEEE Trans. Pattern Analysis Machine Intell., 3(8), 841-846.

[43] Zhao, Q., Fränti, P. (2014). *WB-index: A sum-of-squares based index for cluster validity.* Data & Knowledge Engineering, 92.

[44] Zimmermann, H.J.(1991). *Fuzzy Set Theory and its applications.* Kluwer Academic Publishers, Boston: MA.

APÉNDICE A

Extracción de texturas

A.1. Imagen subgrupo A

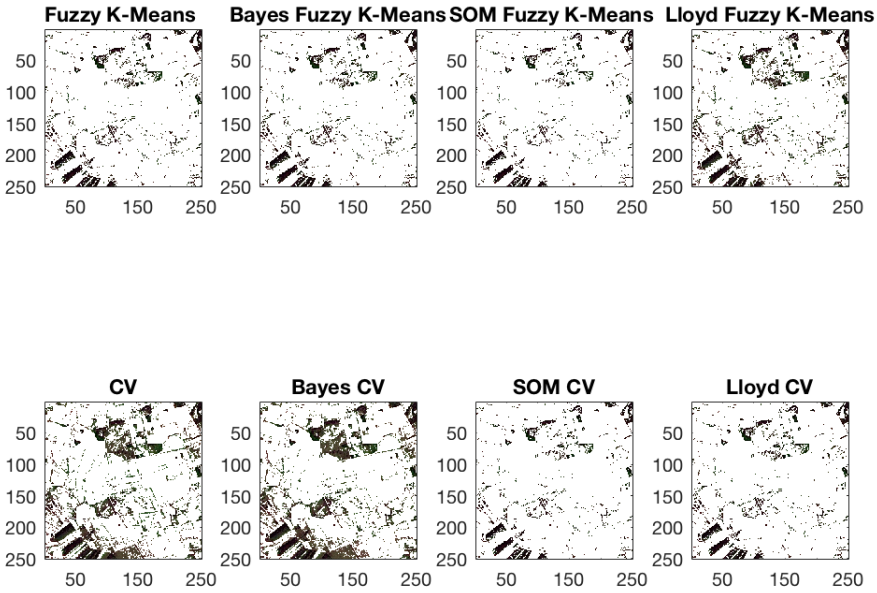


Figura A.1: Texturas clase 1 imagen subgrupo A

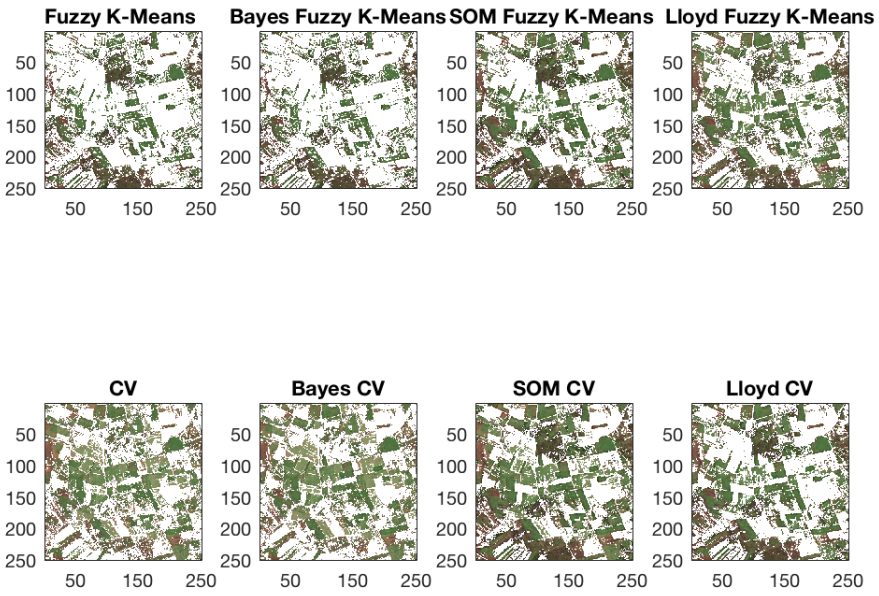


Figura A.2: Texturas clase 2 imagen subgrupo A

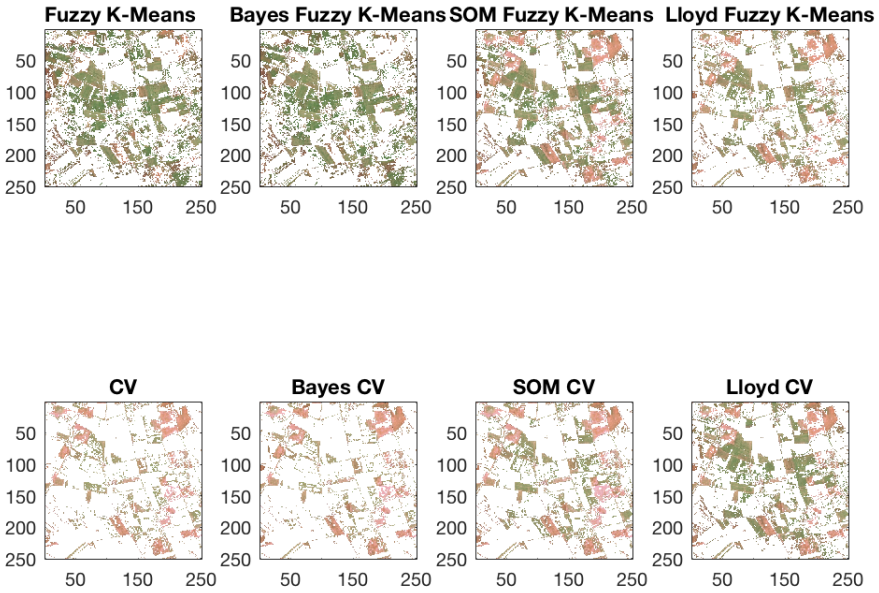


Figura A.3: Texturas clase 3 imagen subgrupo A

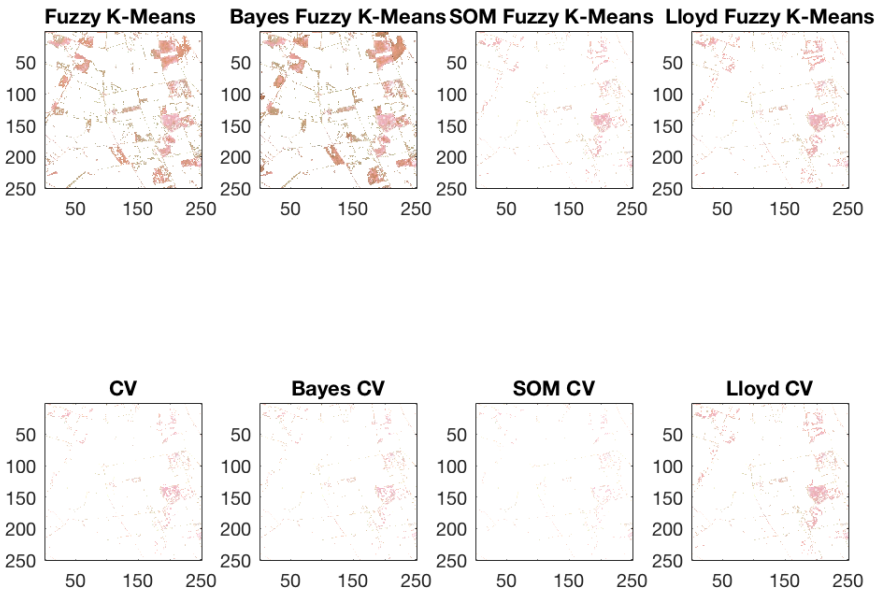


Figura A.4: Texturas clase 4 imagen subgrupo A

A.2. Imagen subgrupo B

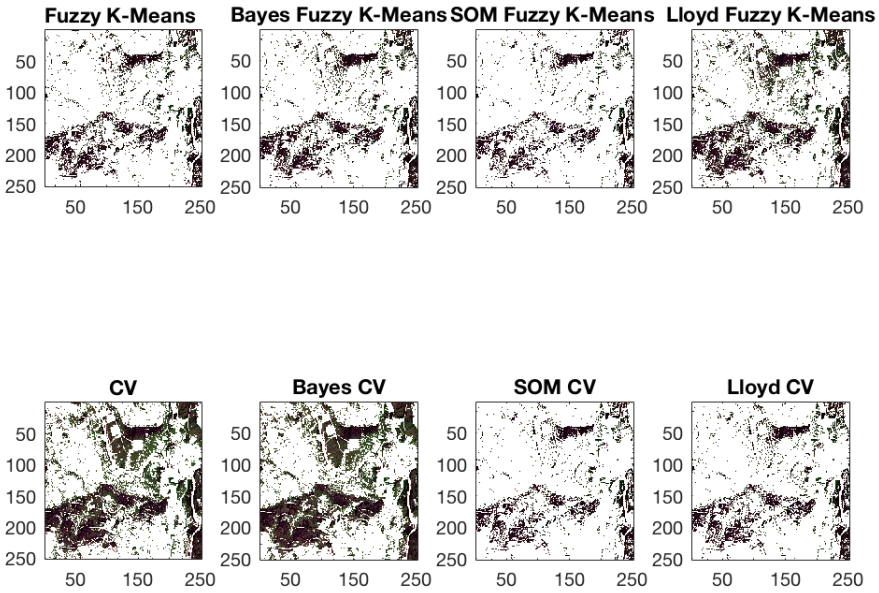


Figura A.5: Texturas clase 1 imagen subgrupo B

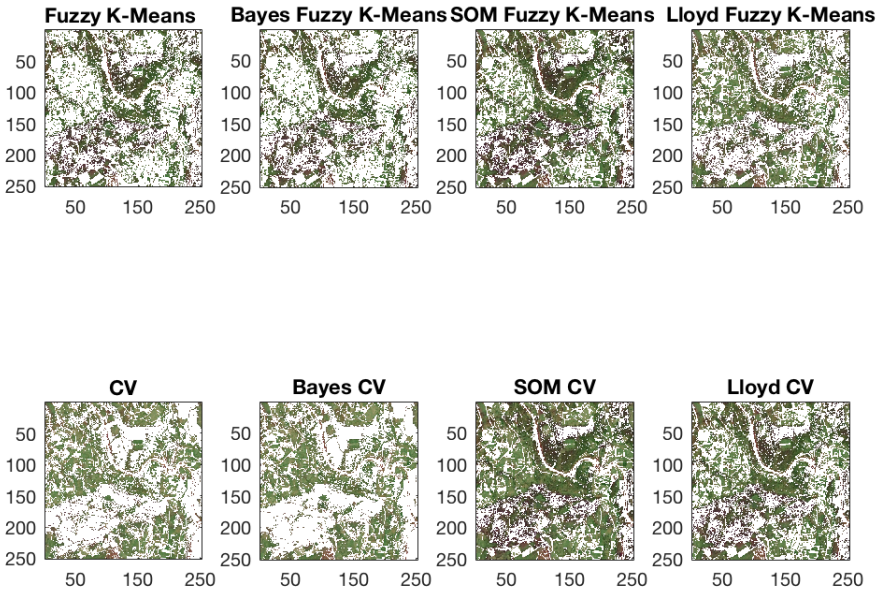


Figura A.6: Texturas clase 2 imagen subgrupo B

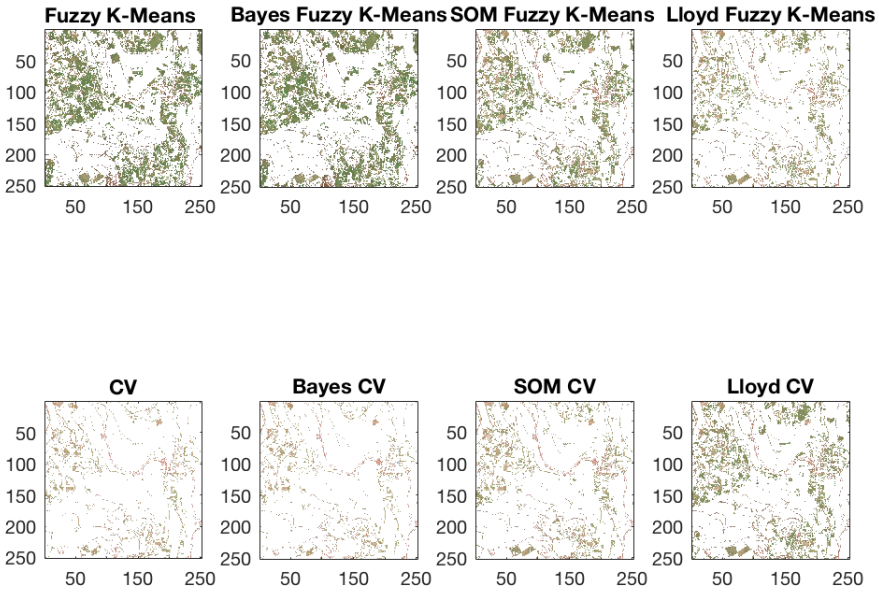


Figura A.7: Texturas clase 3 imagen subgrupo B

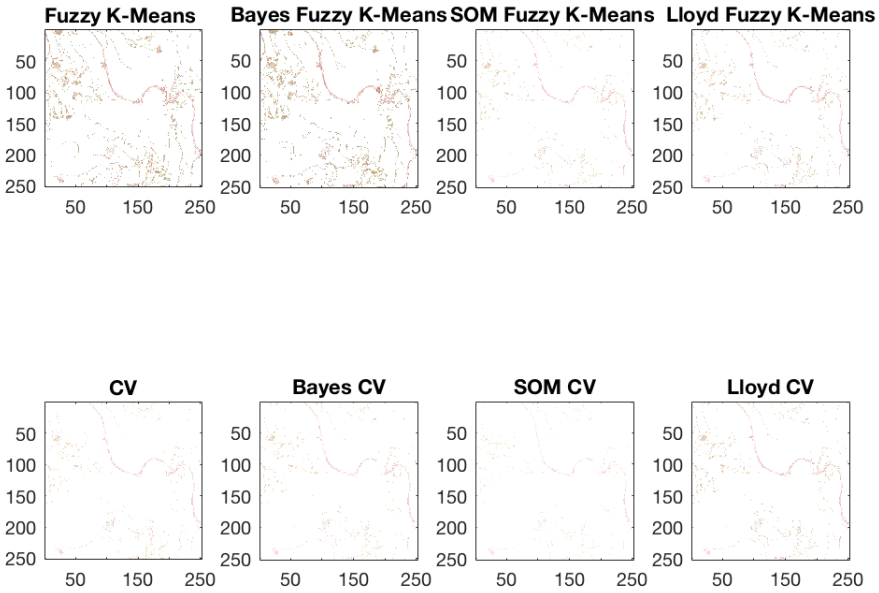


Figura A.8: Texturas clase 4 imagen subgrupo B

APÉNDICE B

Código utilizado

Para la realización de las ejecuciones y la obtención de resultados se ha utilizado un programa implementado en Matlab cuya clase principal permite configurar los parámetros requeridos por cada algoritmo y las fases que se desean llevar a cabo durante la ejecución. En la figura B.1 se muestra la jerarquía de directorios que albergan los archivos de código fuente y el programa principal de la herramienta.

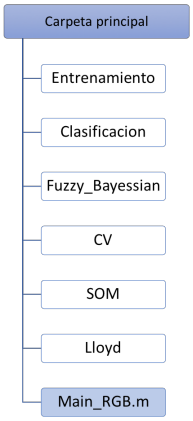


Figura B.1: Jerarquía de directorios código fuente

El directorio *Aprendizaje* contiene el conjunto de imágenes que se utilizan para el entrenamiento de los algoritmos en la fase de aprendizaje. En el directorio *Clasificación* se encuentran todas las imágenes disponibles para ser utilizadas en la fase de clasificación. Los restantes directorios contienen los ficheros Matlab con la implantación de cada algoritmo además de diversas funciones para el análisis y visualización de resultados. Finalmente, en la raiz de la jerarquía se encuentra la clase principal de la herramienta denominada *Main_RGB.m*.

La clase principal permite personalizar los valores de los parámetros requeridos por los algoritmos y otros parámetros generales para el control de las ejecuciones y fases a llevar a cabo. Algunos de los parámetros más relevantes son los siguientes:

- **NOMBRE_BASE_CONOCIMIENTO:** nombre del fichero donde se almacenan todos los datos obtenidos durante la fase de aprendizaje y que posteriormente son utilizados para la clasificación.
- **GUARDAR_PARAMETROS:** guarda los parámetros de todos los algoritmos en un fichero para poder ser utilizados en ejecuciones posteriores.
- **CARGAR_PARAMETROS:** carga los valores de los parámetros previamente guardados en ejecuciones previas, lo que permite tener varias configuraciones de parámetros diferentes para la realización de pruebas.

- **NOMBRE_FICHERO_PARAMETROS:** nombre del fichero que almacena en disco los parámetros de la ejecución a llevar a cabo.
- **APRENDER_<*algoritmo*>:** donde <*algoritmo*> es el nombre del algoritmo a ejecutar en la fase de aprendizaje, por ejemplo APRENDER_LLOYD_FUZZY. Toma los valores $0, 1$.
- **CLASIFICAR_<*algoritmo*>:** donde <*algoritmo*> es el nombre del algoritmo a ejecutar en la fase de clasificación, por ejemplo CLASIFICAR_LLOYD_FUZZY. Toma los valores $0, 1$.

Una vez establecidos los valores de los parámetros deseados, para lanzar la herramienta basta con ejecutar el código del fichero *Main_RGB.m*.