eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

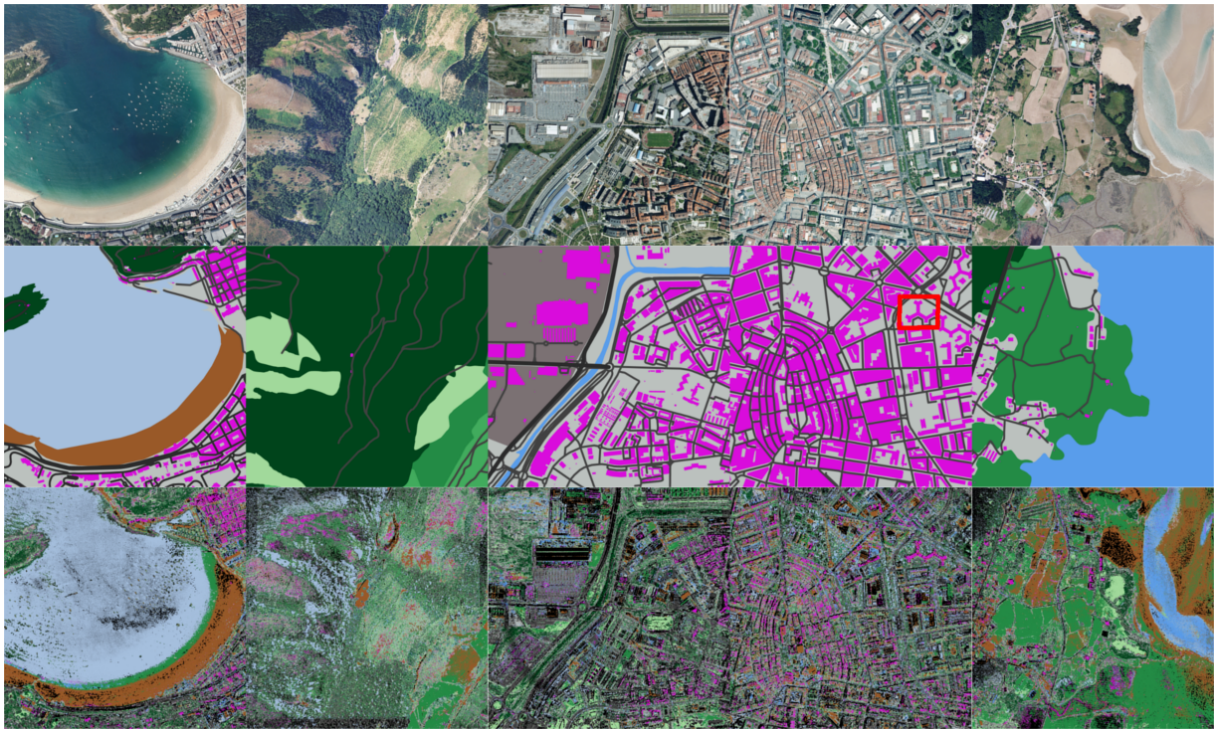
Departamento de Sistemas y Automática

# APRENDIZAJE SUPERVISADO EFICIENTE PARA EL ANÁLISIS DE DATOS GEOESPACIALES A GRAN ESCALA

Memoria presentada para la obtención del grado de Doctor por

JAVIER LOZANO SILVA

Dirigida por:
Prof. Dr Ekaitz Zulueta
Dr. Marco Quartulli



*A Elene, Olaia, Miren, Ricardo, Marisol y Elena,*
*por aguantarme.*
*(sí, en ingles hubiera estado mejor, lo sé)*

# Agradecimientos

Son muchas las personas a las que debo el haber llegado hasta aquí y de las que espero acordarme y no dejarme a nadie en el camino.

En primer lugar, al Profesor Ekaitz Zulueta por su dedicación, entrega, ayuda y demás durante todo el proceso doctoral. Una travesía que, tras un comienzo fulgurante, atravesó un duro desierto durante unos años, para finalizar ahora. Y, en todo ese tiempo, buscando caminos, senderos y atajos que nos sacaran de allí. Benetan, mila esker.

En segundo lugar, al Doctor Marco Quartulli, cuya aparición fue providencial para abrir un pequeño sendero hacia el mundo de la teledetección y poder salir del desierto en el que nos encontrábamos. Agradecer su inestimable ayuda en el día a día, por los miles de consejos, opiniones, ideas, discusiones, ánimos, etc. Por convertir los problemas en oportunidades de mejora. También, por decidirse por un lenguaje de programación y hacer poesía. Mila esker Q.

Agradecer también a Igor G. Olaizola su ayuda, tiempo y consejos pero, sobretodo, la compañía prestada en este proceso doctoral, tras los percances sufridos en la, ahora lejana, travesía por el desierto.

Gracias a Naiara por compartir alegrías y penas en esto del *"proceso doctoral"*, ¡ánimo que ya estamos!

No quisiera olvidarme del Profesor Pedro Iriondo Bengoa, por aceptarme como alumno de doctorado, por su disponibilidad para ayudarme y aconsejarme en cualquier momento, antes, durante y después de ser mi tutor.

No puedo olvidarme de los compañeros del departamento de Televisión y Servicios Multimedia de Vicomtech, siempre dispuestos a ayudar. Mila esker a todos. A los que están (Felipe, Iosu, Angel, Aritz, Inaki, Ana, Mikel, Iñigo, Ion y Iosu). Y a los

que no están (Julen, Maider, Kevin y Mikel).

Tampoco puedo olvidarme de esa gente que hace que levantarse para ir a trabajar, sea más fácil. ¡¡¡Tiburones!!! Mila esker eta ez aldatu.

Gracias a la cuadrilla, por hacerme desconectar un rato de todo esto y por estar siempre ahí.

A mi familia, ya que sin su apoyo y ayuda esto hubiera sido imposible. Porque sois una parte del equipo que me hace falta. A mis padres, por estar siempre ahí, ayudarme cuando me hacía falta y cuando no. Y por darme la oportunidad de elegir. A Elena, por aguantarme. Lo siento, es el precio que hay que pagar por ser la hermana pequeña.

Por último, a Miren, Elene y Olaia. Porque hacéis que el mundo gire, porque con una sonrisa vuestra se acabaron los problemas, porque sois la otra parte del equipo. Mila esker.

# Resumen

El presente trabajo de tesis doctoral tiene como objetivo comprobar la viabilidad de la integración de funcionalidades de aprendizaje automático en servidores de mapas web. La validación de esta hipótesis se ha realizado mediante su implementación en un prototipo pre-operacional. Esta implementación ha consistido en el desarrollo de una plataforma para el mapeo temático sobre imágenes de teledetección de muy alta resolución mediante aprendizaje supervisado a través de una plataforma web. Integrando las capacidades de escalabilidad de los modernos algoritmos de aprendizaje automático y las de los servidores de mapas web, la hipótesis supera el estado del arte actual, caracterizado por la separación de los dos ámbitos que requiere la continua aportación del experto de teledetección en tareas de mapeo temático intensivo. Mediante esta aportación, se abre el campo aplicativo referido a la creación semi-automática de mapas temáticos dedicados y a gran escala en diferentes ámbitos. Estos van desde la agricultura hasta la monitorización medioambiental, por parte de usuarios expertos de dichos dominios aplicativos y sin conocimientos específicos sobre técnicas de teledetección. Dicho desarrollo se fundamenta en facilitar la explotación de datos de teledetección mediante plataformas de aprendizaje automático de fácil acceso que aumenten las capacidades de análisis de datos, de forma que los campos aplicativos puedan expandirse. Estas capacidades pueden concretarse en algoritmos de etiquetado semántico basados en métodos de clasificación supervisada, de forma que un mapa temático pueda ser generado a partir de datos raster adquiridos por sistemas de teledetección y en función de las necesidades del usuario. Para ello, es necesaria la integración de capacidades de aprendizaje automático dentro del servidor de mapas web, junto con una interfaz sencilla que permita la navegación geoespacial y la supervisión del aprendizaje. El carácter adaptativo del aprendizaje, junto con

su integración en un servidor web, requiere un algoritmo de clasificación con una gestión y procesamiento de datos eficiente en términos de tiempo de procesamiento compatibles con la navegación web tradicional. Al mismo tiempo, el volumen de datos gestionado por aplicaciones de teledetección motiva el traslado de la metodología a entornos en la nube bajo el paradigma *Big Data*.

# Abstract

The present thesis aims to test the viability of the integration of machine learning capabilities into web map servers. The validation of this hypothesis has been carried out by the development of a pre-operational prototype. The developed prototype is a platform for thematic mapping by supervised learning from very high resolution remote sensing imagery data through a web platform. This contribution overcomes the current state of art, characterized by the separation of the two areas, which requires a continuous involvement of remote sensing experts in thematic mapping intensive tasks: labour intensive tasks are supplemented by the integration of the scalability capabilities from machine learning engines and web map servers. With this hypothesis the application field referred to the semi-automatic creation of large scale thematic maps can open up different fields, from agriculture to the environmental monitoring field, to expert users of these applications domains with limited specific knowledge of remote sensing techniques. Semantic tagging algorithms based on supervised classification methods can be exploited for thematic map creation from raster data based on user needs. This requires the integration of machine learning capabilities within web map servers, along with a simple interface that enables navigation and the monitoring of geospatial learning. The adaptive nature of this learning, along with its integration into a web server, requires a classification algorithm characterized by efficient management and processing of data in time scales compatible with traditional web browsing. At the same time, the volume of data managed by remote sensing applications motivates the transfer of the developed methodology to cloud environments under the Big Data paradigm.

# Índice general

# Índice de figuras

# Lista de Tablas

# Lista de Abreviaturas

EO Observación de la Tierra o *EO*, de las siglas en inglés *Earth Observation*.

GRSS Geoscience & Remote Sensing Society.

GIS Sistema de información geográfica, en ingles *Geographic information system*.

Raster Datos sin procesar, en este caso imágenes sin procesar.

AVIRIS (Airborne Visible/Infrared Imaging Spectrometer)

KDD Descubrimiento en conocimiento en bases de datos, de las siglas en inglés de *Knowledge Discovery in Databases*.

OLTP Procesamiento de Transacciones En Línea, de las siglas en inglés de *OnLine Transaction Processing*. Tipo de procesamiento.

CBIR Consulta de imágenes mediante ejemplo, de las siglas en inglés *Content-based Image Retrieval*.

HOG Descriptor *Histogramas de gradientes orientados*, de las siglas en inglés *Histograms of Oriented Gradient*.

LBP Descriptor *Patrón Local Binario*, de las siglas en inglés *Local line Binary Pattern*.

LSD Descriptor *Detector de segmentos de linea* de las siglas en inglés *Line Segment Detector*

tp Verdaderos positivos, de las siglas en ingles *true positive*

| | |
|---|---|
| tn | Verdaderos negativos, de las siglas en ingles *true negative* |
| fp | Falsos positivos, de las siglas en ingles *false positive* |
| fn | Falsos negativos, de las siglas en ingles *false negative* |
| pdf | Función densidad de probabilidad *probability density function* |
| cdf | Función de distribución acumulada *Cumulative distribution function* |
| árbol $k$-d | Árbol $k$-dimensional |
| $k$-NN | $k$ vecinos más cercanos, de las siglas en ingles *$k$-Nearest Neighbor* |
| $k$-PNN | Consulta con Probabilidad de $k$ vecinos más cercanos, de las siglas en ingles *Probabilistic $k$-Nearest-Neighbor Query* |
| T-$k$-PNN | Consulta con Umbral de Probabilidad $k$ vecinos más cercanos, de las siglas en ingles *Probabilistic Threshold $k$-Nearest-Neighbor Query* |
| API | interfaz de programación de aplicaciones *Application Programming Interface* |
| FPGA | Circuito integrado configurable *Field Programmable Gate Array* |
| GPU | Unidad de procesamiento gráfico *Graphics Processing Unit* |
| RDD | Abstracción para la distribución de datos en Spark, del inglés *Resilient Distributed Dataset* |
| svg | Gráficos Vectoriales Redimensionables, del inglés *Scalable Vector Graphics*. Especificación para describir gráficos vectoriales. |
| css | Hoja de estilo en cascada, del inglés *cascading style sheets*. Lenguaje para definir la apariencia de un archivo HTML. |

# Lista de Símbolos

$S$ Clase objetivo.

$D$ Base de datos con incertidumbre o regiones de imágenes a clasificar.

$k$ Número de puntos.

$p(S)$ Cuantificación de probabilidad de $S$.

$T$ Umbral de probabilidad.

$o_i$ Objetos con incertidumbre $i$ de $D(i = 1, ..., |D|)$ o regiones de entrenamiento.

$q$ Objeto de consulta.

$q$ Región de imagen con clase desconocida.

$r_i$ $|o_i - q|$ Distancia entre el objeto de consulta y el objeto con incertidumbre.

$d_i(r)$ $PDF$ de $r_i$ (Distancia Función Densidad de Probabilidad).

$D_i(r)$ $CDF$ de $r_i$ (Distancia función de distribución acumulada) o distancia relativa a otro objeto de la clase.

$D_h(r)$ $CDF$ de $r_h$ (Distancia función de distribución acumulada) o distancia relativa hacia otras clases.

$x$ Muestra desconocida.

$v$ Valor de corte, mediana de las coordenadas discriminantes.

$c$ Coordenada de $x$ discriminante para ese nodo.

$d_{nn}$ Distancia al vecino más cercano.

# Capítulo 1

# Introducción

## 1.1. Motivación

Los sistemas de minería de datos orientados a la Observación de la Tierra o *EO*, de las siglas en inglés *Earth Observation*, son objeto de continuas investigaciones y desarrollos por parte de la comunidad científica [1–3], como muestra la Figura 1.1. El volumen de los archivos de datos, a escala de Petabytes, crece a un ritmo de 10 Gigabytes por día, un contenido al que no se ha accedido en su mayoría [1].

Los datos *raster* aeroespaciales, como las imágenes de teledetección, son a menudo difíciles de interpretar, por lo que es necesario crear mapas temáticos que transformen ese contenido en información interpretable por el usuario final. La incertidumbre que caracteriza este tipo de datos [4] justifica la utilización de enfoques probabilísticos para las tareas de búsqueda, modelado y análisis, que puedan ayudar en la tarea de comprensión de los datos. A nivel aplicativo, el aprendizaje automático puede proporcionar herramientas para la búsqueda, el análisis y la modelización, capaces de enfrentarse a datos de alta complejidad y con un elevado grado de incertidumbre.

El aumento de capacidades para el análisis de metadatos y contenidos puede suponer una potencial expansión del análisis y explotación de datos de *EO*, desde usos científicos y profesionales de teledetección hasta expertos en tecnologías de aplicación en múltiples sectores. Los posibles campos de aplicación del trabajo presentado son muy diversos, a tenor de los resultados obtenidos en la búsqueda de aplicaciones relacionadas con la teledetección dentro de la plataforma *Web Of Science* de Thomson

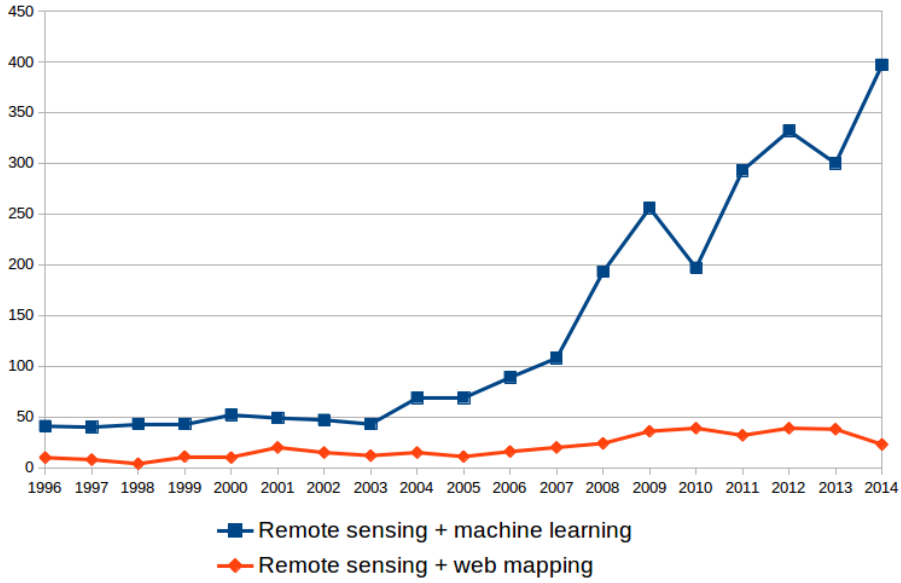



Figura 1.1: Publicaciones por año registradas en la *Web of Science* sobre aprendizaje automático y web mapping en teledetección

Reuters [1]. Entre los resultados obtenidos, se pueden destacar las siguientes categorías de aplicación: agricultura, servicios forestales, geología, hidrología, detección de grandes masas de hielo, tipo y uso de la cobertura terrestre, monitorización de mares, océanos y costas, y medición de parámetros meteorológicos.

Todo esto facilita una transición aplicativa, representada en una evolución desde la cartografía web para usuarios ocasionales, hasta la creación de mapas temáticos de cobertura basados en las necesidades especificadas interactivamente por expertos de diferentes dominios.

La base para este tipo de herramientas puede representarse mediante algoritmos de etiquetado semántico basados en métodos de aprendizaje automático de clasificación supervisada. Idealmente, un mapa temático puede ser generado a partir de datos *raster* y en base a los objetivos específicos del usuario, en lugar de ser más genérico y generado por complicados y laboriosos procesos manuales. Alcanzar la automatización de la creación de un mapa temático a partir de datos *raster* requiere

[1] http://wokinfo.com/

la integración de las capacidades de aprendizaje automático dentro del servidor de mapas. Para ello, se considera la creación de un sistema de clasificación supervisado, integrado en una arquitectura servidor web escalable que implemente interfaces sencillas que combinen la navegación geoespacial y el componente de supervisión. Esto implica un conjunto de consideraciones para que un sistema aplique este concepto: *usabilidad* [5], *escalabilidad* [6] y *efectividad* [7].

La clasificación debe poder modelar el entrenamiento para cada caso de uso particular según las necesidades propias del usuario en cada momento, lo que requiere la utilización de un modelo algorítmico de carácter adaptativo. Considerando el tiempo máximo de espera para la descarga de una página web en 40 segundos [8], el tiempo de procesamiento adquiere un carácter mucho más relevante que en las implementaciones clásicas de aprendizaje automático. En éstas, el tiempo de procesamiento no está tan condicionado y la duración de los procesos con resultados aceptables varía desde pocos minutos a horas [9]. Es por ello que se requiere una gestión y procesamiento eficiente de los datos donde la utilización de estructuras de indexación que agilicen el procesamiento de los datos resulta muy relevante.

Tal y como se comentaba al comienzo de la sección, con las tecnologías actuales se pueden obtener imágenes de la Tierra con resolución métrica, con lo que nos podemos encontrar con Terabytes o incluso Petabytes de datos. Si se requiere procesar todo este volumen de datos es necesario trasladar la implementación a entornos *Big Data*. En este aspecto, debido al repetido acceso a los datos en memoria por parte de los algoritmos de clasificación, se debe considerar el uso de infraestructuras que permitan el acceso repetido a memoria.

## 1.2. Objetivos

El presente trabajo de tesis doctoral tiene como objetivo comprobar la viabilidad de la integración de funcionalidades de aprendizaje automático en servidores de mapas web. Integrando las capacidades de escalabilidad de ambos, la hipótesis supera el estado del arte actual, caracterizado por la separación de los dos ámbitos que requieren la continua intervención del experto de teledetección en tareas de mapeo temático intensivo. Mediante esta hipótesis, se abre el campo aplicativo a la

posibilidad de creación de mapas temáticos dedicados en diferentes ámbitos, que van desde la agricultura hasta el campo medioambiental.

La validación de la hipótesis presentada se ha realizado mediante su implementación en un prototipo pre-operacional. Esta implementación ha consistido en el desarrollo de una plataforma para el mapeo de datos temáticos sobre imágenes de teledetección de muy alta resolución mediante aprendizaje supervisado a través de una plataforma web.

Este objetivo requiere integrar funcionalidades de aprendizaje automático en servidores web y habilitar procesos de supervisión capaces de modelar la incertidumbre derivada de la supervisión, mediante procesos probabilísticos a través de interfaces web. Procesar los volúmenes de datos típicos en teledetección a través de plataformas web requiere de sistemas escalables que permitan obtener resultados casi en tiempo real.

De este modo, se cubren cuatro de las siete "V"-s que definen el paradigma *Big Data* [10] volumen, velocidad, variedad, veracidad, validez, veracidad, volatilidad y valor.

**Volumen:** analizar datos provenientes del ámbito de la teledetección, generalmente grandes volúmenes de datos. Abarca la primera parte de la definición.

**Velocidad:** relacionada con la cantidad de datos que se generan al día. En este concepto se une la velocidad de procesamiento necesaria para procesar grandes volúmenes de datos casi en tiempo real, con el tiempo de espera habitual para paginas web.

**Veracidad:** relacionada con la incertidumbre de los datos. Es cubierta por algoritmos de clasificación supervisada desarrollados para ello.

**Valor:** es el resultado que se desea obtener después de analizar los datos. En este aspecto, la visualización de los mismos puede ayudar a valorar este aspecto.

La visualización de los datos contempla la presentación de los datos de manera comprensible y accesible [11], en este caso mediante la presentación en forma de mapas de los resultados clasificados a través de una interfaz web. Alcanzar este objetivo requiere el desarrollo e implementación en diferentes áreas temáticas como

el aprendizaje automático, el desarrollo web o la programación sobre el paradigma *Big Data*. Para ello, se han abordado los siguientes objetivos específicos:

- Implementación eficiente de algoritmo para clasificación supervisada sobre imágenes de teledetección de muy alta resolución.
- Diseño e implementación de arquitectura para facilitar la accesibilidad del prototipo mediante plataforma web.
- Creación de una interfaz de usuario que permita la configuración de los algoritmos implementados, con cierto grado; y la selección de datos para el modelado de los mismos.
- Estudio de aproximación del objetivo principal al análisis de grandes volúmenes de datos en entornos *Big Data*.
- Evaluar el sistema mediante resultados cuantitativos, sobre el rendimiento de la clasificación y sobre el tiempo de procesamiento.
- Desarrollo de una plataforma para la creación de mapas de validación basada en datos abiertos *open data*, con los que evaluar el sistema en términos de rendimiento de clasificación.

Figura 1.2: Imagen resumen sobre los resultados obtenidos en la tesis. El sistema de mapeo temático vía web desarrollado es evaluado sobre 5 zonas diferentes.

# Capítulo 2

# Fundamentos del aprendizaje automático

En este capítulo se realiza una breve descripción de los fundamentos del aprendizaje automático, con el fin de establecer los criterios principales de este trabajo y facilitar la comprensión del documento. Se ha considerado conveniente incluir diferentes aspectos que complementan el trabajo desarrollado, desde la caracterización de los datos hasta las metodologías para hacer frente al procesamiento de grandes volúmenes de datos.

## 2.1. Descubrimiento de conocimiento en bases de datos

Fayyad define el KDD en [12] como "el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de datos". En esta definición se indican cuáles deben ser las propiedades del conocimiento extraído:

- Válido: los patrones adquiridos deben de ser precisos, con cierto grado de incertidumbre, con nuevos datos.
- Novedoso: debe aportar información previamente desconocida al sistema y, sobre todo, al usuario.

- Potencialmente útil: la información obtenida debe servir para obtener algún beneficio.

- Comprensible: la obtención de patrones poco comprensibles dificulta su interpretación, la extracción de conocimiento y su uso para la toma de decisiones.

Tal y como se puede deducir de la definición anterior, el KDD es un proceso complejo que incluye, además de la obtención de patrones que es el objetivo específico del *data mining*, la preparación de los datos, la evaluación de los datos para comprobar la calidad y la evaluación de los patrones y la interpretación de los mismos, tal y como muestra en la Figura 2.1.

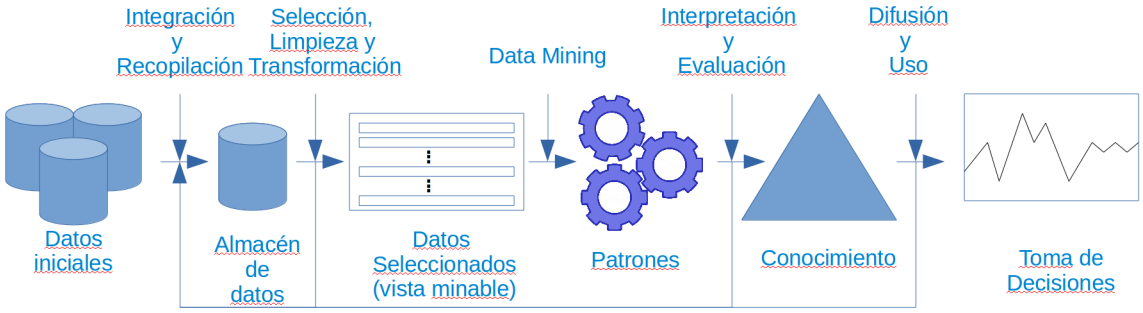



Figura 2.1: Esquema proceso KDD. Fases de gestión, análisis y explotación de datos.[12]

Por otro lado, esta definición deja clara la relación entre el KDD y la minería de datos. Mientras que el KDD es el proceso que engloba los procesos para descubrir conocimiento útil en bases de datos, la minería de datos se encarga específicamente de generar o encontrar patrones o modelos.

Como se puede ver en la Figura 2.1, el proceso de KDD está compuesto por diferentes etapas que analizaremos a continuación.

En la primera fase *Integración y recopilación* se determinan las fuentes de información útiles. A continuación, se unifican los datos dándoles un formato común, transformándolos y normalizándolos para un mejor rendimiento en los procesos de memoria, que facilita la navegación y visualización entre datos. Se puede dar el caso de que los datos necesarios estén dispersos en diferentes bases de datos, lo que resulta

un problema debido a la utilización de diferentes formatos de registro. Por ello, se requiere integrar todos los datos que se van a analizar.

A continuación, se determina la calidad de los datos a analizar, en la fase de *Selección, limpieza y transformación*. En esta fase, se seleccionan y preparan los datos para su posterior clasificación, ya que no todos los datos son realmente relevantes o necesarios.

Otra tarea de esta fase es la preparación de los datos. Se pueden generar nuevos atributos aplicando alguna función a los atributos originales, o modificando los mismos atributos o discretizándolos, aplicando un análisis de componentes principales,... según sea el caso. Como se puede observar, la relevancia de los atributos es de vital importancia para obtener buenos resultados.

La tercera fase, la fase de *Data Mining*, es la más característica del proceso de KDD, por lo que es común utilizar este término para referirse a todo el proceso. Como hemos mencionado anteriormente, esta fase es la encargada de producir y proporcionar nuevo conocimiento a través de un modelo basado en los datos proporcionados. El modelo es una descripción de relaciones entre los datos, que puede usarse para hacer predicciones o entender mejor los datos. Para obtener el modelo, es necesario tomar las siguientes decisiones:

- Determinar la tarea de minería (Clasificación, clustering u otros)
- Elegir el tipo de modelo: (Evaluación de similaridad, árboles de decisión, redes neuronales,...).
- Elegir el algoritmo de minería. ($k$-nn, J48, Id3, NBTree, ..., Perceptron Multicapa, Redes Neuronales Autoorganizadas)

Las tareas que se pueden encontrar en la fase de *Data Mining* se pueden separar en dos clases: predictivas y descriptivas. En el grupo de las predictivas están la clasificación y la regresión. En las descriptivas nos encontramos con el agrupamiento -en inglés *clustering*-, reglas de asociación, reglas de asociación secuenciales y correlaciones.

**Clasificación** En esta tarea, cada instancia o registro de la base de datos pertenece a una clase que se le asigna a través del atributo discreto clase de instancia.

El objetivo es predecir la clase de nuevas instancias de las que se desconoce la clase.

**Regresión** La principal diferencia con la clasificación es que, en la regresión, se asigna un valor real a cada instancia a través del aprendizaje de una función. El objetivo es predecir el número que hay que asignar, minimizando el error entre el valor predicho y el valor real.

**Agrupamiento o *Clustering*** Se trata de una tarea descriptiva, que consiste en obtener grupos naturales a partir de los datos. La diferencia con la clasificación y la regresión es que analiza los datos para crear la clase. Se forman grupos entre datos similares entre sí, y diferentes con los datos de los otros grupos.

**Correlaciones** Buscan el grado de similitud entre diferentes valores de instancias.

**Reglas de Asociación** Similar a las correlaciones, tiene como objetivo buscar relaciones no explícitas entre atributos categóricos.

El trabajo presentado en esta tesis se ha enfocado desde la tarea de clasificación para llevar a cabo las implementaciones presentadas en las siguientes secciones. Se trata de una tarea englobada dentro de los métodos de *Aprendizaje Supervisado*, también conocidos como *Clasificación Supervisada*.

A la hora de construir el modelo, los datos se distribuyen en tres grupos diferentes: el primero para el entrenamiento, el segundo para testear y el tercero para validar el modelo. En la fase de entrenamiento se construye el modelo algorítmico. En la fase de test se ajustan los parámetros del algoritmo para obtener un mejor resultado. En la fase de validación se comprueba que los ajustes realizados mejoran los resultados, generalizando la solución al problema sin provocar un sobreentrenamiento que sea demasiado específico sobre una parte de los datos. De esta manera, nos aseguramos de que el algoritmo realmente ha aprendido, ya que los datos de validación no son conocidos por el modelo, logrando unas predicciones más fiables y robustas. La distribución de los datos debe realizarse con sumo cuidado, ya que si el volumen de datos de entrenamiento es pequeño, el algoritmo no modelará correctamente el problema y, si es demasiado grande, se sobreentrenará, realizando

predicciones demasiado específicas, efecto conocido como sobreajuste, o *overfitting* en inglés.

La cuarta fase se conoce como *Interpretación y evaluación*. En esta fase se evalúan y analizan los patrones generados. En caso de ser necesario, puede volverse hacia atrás para realizar algún ajuste en alguna fase anterior. Los patrones obtenidos deben ser, idealmente, ajustados a los requerimientos de precisión, robustez, claridad y rapidez establecidos para caso de uso.

Para realizar la validación de los modelos generados, las técnicas básicas son la validación simple y la validación cruzada con $n$ pliegues, en inglés *n-fold cross validation*.

La validación simple se basa en separar el conjunto de datos en dos subconjuntos, tal y como se indicaba en la fase de construcción del modelo, pudiendo variar el porcentaje de datos de validación, típicamente entre el 5 y el 50 por ciento [13].

La técnica de validación cruzada de $n$ pliegues consiste en separar el grueso de los datos en $n$ grupos, realizando $n$ iteraciones. En cada iteración uno de los grupos es para la validación y, el resto, para crear el modelo. La distribución de los datos en los grupos se hace de forma estratificada para evitar distorsiones, quedando una distribución homogénea de los datos. En este caso, se obtienen $n$ ratios de error y precisión que se promedian para calcular el ratio de error y precisión del modelo final.

Por último, está la fase de *Uso y difusión*. Como su propio nombre indica, una vez construido y validado el modelo, solo queda usarlo, bien para que un analista experto recomiende acciones según los resultados obtenidos, o bien para aplicar diferentes conjuntos de datos al modelo.

## 2.2. Clasificación supervisada

Como ya se ha visto, la *Clasificación Supervisada* tiene como objetivo aprender de referencias categorizadas para asignar una clase a una instancia o conjunto de datos.

La acción de asignar una clase conocida a una instancia lleva, de manera implícita, el enriquecimiento semántico del sistema. Dicho de otro modo, pasamos a tener

una serie de instancias que hacen referencia a un objeto definido y conocido. Esta particularidad obliga a tener un conjunto de instancias categorizadas en la fase de entrenamiento. La distribución de las instancias entre las clases tiene que ser realizada por un supervisor con el conocimiento suficiente para diferenciar las diferentes clases. Otra forma de verlo es que la *Clasificación Supervisada* permite incluir al usuario en el proceso de aprendizaje, de manera que puede reflejar su conocimiento sobre los datos a analizar.

En el caso de las implementaciones presentadas en el capítulo 5, la incorporación del usuario dentro del proceso de aprendizaje es vital, ya que *a priori* no hay clases predefinidas y es el usuario quien decide el objetivo a buscar. Además, gracias a la interacción implementada en el sistema, el usuario es capaz de modificar los datos de entrenamiento propuestos al algoritmo, modificando su comportamiento, según las necesidades de cada caso.

## 2.3. Minería de datos sobre imágenes

El desafío fundamental en la Minería de Imágenes es determinar cómo el bajo nivel (semántico), entendido como la representación de píxeles de una imagen en bruto o de una secuencia de imágenes, puede ser procesado para identificar objetos y relaciones a alto nivel [14]. La extracción de conocimiento acerca de la imagen partiendo de características se conoce como Brecha Semántica o *"Sematic Gap"* en inglés.

Por definición [14], la Minería de Imágenes se refiere a la extracción de patrones de imágenes a partir de bases de datos de imágenes. La Minería de Imágenes es diferente a la visión por computador a bajo nivel y de las técnicas de procesamiento de imágenes, ya que el centro de atención de la Minería de Imágenes está en la extracción de patrones de una colección de imágenes, tratando de superar el *Semantic Gap*. En la Minería de Imágenes, el objetivo es el descubrimiento de patrones que son representativos en una determinada colección de imágenes.

## 2.4. Algoritmo *k* vecinos más cercanos

Una forma básica de clasificar un caso es asignarle la misma clase que a otro caso similar cuya clasificación es conocida. Entre este tipo de métodos destacan los métodos de clasificación por el vecino más cercano o *k-NN*, de las siglas en inglés *k-Nearest Neighbor* [15].

### 2.4.1. Características del clasificador *k-NN*

El método de clasificación de los $k$ vecinos más cercanos ($k$-$NN$) es uno de los métodos de clasificación más sencillos y, a la vez, de los más potentes que existen [16]. Un clasificador $k$-$NN$ destaca por su robustez frente al ruido y por tener una buena capacidad de aproximación.

Entre sus ventajas destacan la sencillez de implementación y ajuste, ya que solo dispone de dos parámetros: $k$, el número de vecinos más cercanos a considerar; y $r$, el radio de vecindad y la robustez. Por otro lado, el modelo generado puede ser actualizado *on-line* con nuevos casos de las clases ya definidas.

Sin embargo, para que su funcionamiento sea bueno, el espacio de características debe estar lo suficientemente poblado de ejemplos (para que tenga sentido clasificar un ejemplo dado, en función de los $k$ ejemplos más cercanos). Por desgracia, esto es más difícil de conseguir cuando la dimensionalidad de los datos aumenta. En este caso, es interesante la utilización de técnicas de reducción de dimensionalidad (p.e: componentes principales [17]).

Por otro lado, en el clasificador $k$-$NN$ básico, todos los atributos intervienen por igual, lo que en determinadas ocasiones degrada su comportamiento, ya que se deben calcular las distancias respecto a cada elemento. Para solucionar este problema, se puede ponderar la aportación de cada atributo o bien, como en el caso anterior, utilizando algún método de extracción de características.

Otra de sus desventajas es la sensibilidad con la que, debido a atributos ruidosos o irrelevantes, el cálculo de la distancia se ve afectado. El número de muestras por clase utilizado también afecta fuertemente a la sensibilidad. Este último problema se puede resolver mediante el muestreo equilibrado entre las diferentes clases.

### 2.4.2. Método de clasificación del vecino más próximo

Para clasificar un nuevo elemento mediante el algoritmo $k$-NN se debe hacer lo siguiente:

- Determinar el radio de vecindad y el valor de $k$
- Presentar el elemento a clasificar en el espacio de características multidimensional
- "Trazar" una hiperesfera con centro en el elemento de consulta. La hiperesfera deberá contener, como mínimo, algún elemento. De lo contrario, se debe modificar el radio de vecindad determinado
- Asignar la clase al elemento que se va a clasificar, en función del valor $k$ y el número de elementos dentro de la hiperesfera.

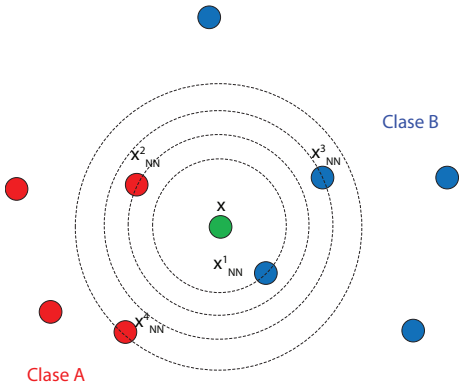



Figura 2.2: Clasificación $k$-NN, diferencia de radios de vecindad

En la Figura 2.2, se presenta un ejemplo con diferentes radios de vecindad. Dependiendo del radio de vecindad seleccionado, el número de $k$ vecinos y la métrica de distancia seleccionada, la clasificación puede cambiar. Así, considerando únicamente el primer radio, el elemento será clasificado como $B$. Mientras que si seleccionamos el cuarto radio, al ser igual el número de elementos por clase, se tendrían que calcular las distancias a los diferentes elementos para obtener la clase.

**Algoritmo 1** Algoritmo $k$-*NN* Básico

**Entrada:** $D = (x_1, c_1), ..., (x_N, c_N)$
$x = (x_1, ..., x_n)$ (nuevo caso a clasificar)
**Salida:** Datos Clasificados.
**para** todo objeto ya clasificado $(x_i, c_i)$ **hacer**
Calcular $d_i = d(x_i, x)$
Ordenar $d_i (i = 1, ..., N)$ en orden ascendente
Quedarnos con los $K$ casos $D_x^K$ ya clasificados más cercanos a $x$
Asignar a $x$ la clase más frecuente en $D_x^K$
**fin para**

### 2.4.3. *T-$k$-PNN en bases de datos*

En aplicaciones como los servicios de localización GPS o en aplicaciones con sensores de monitorización, los valores de los elementos de las bases de datos tienen asociado un grado de incertidumbre asociado a la propia medida.

Para afrontar la creciente necesidad de gestión de datos con incertidumbre y proveer servicios de alta calidad, algunos investigadores han propuesto el uso de "bases de datos con incertidumbre". En particular, estos datos con incertidumbre son evaluados mediante consultas probabilísticas, que generan respuestas probabilísticas y estadísticas [18–21].

Un modelo de datos ampliamente utilizado, asumido por bases de datos con incertidumbre es la incertidumbre del atributo, donde el valor real del atributo es localizado en una zona concreta o en una región de incertidumbre. La Figura 2.3 muestra un ejemplo de esta consideración, en lo que podría ser un servicio de localización, donde la incertidumbre de la localización de un objeto en movimiento puede ser tratada como una distribución Gaussiana normal [22, 23]. La región de incertidumbre es un área circular con un radio conocido como "distancia umbral". Cuando se sobrepasa el umbral marcado en la Figura 2.3, se informa al sistema de la nueva localización.

El algoritmo *T-P-kNN*, considerado como opción fundamental en este trabajo, está basado en un estudio de *k-PNN*, de las siglas en inglés *Probabilistic k-Nearest Neighbor Query* [24], para bases de datos con atributos con incertidumbre. El *k-PNN* puede considerarse la versión del algoritmo k-vecinos más cercanos para evaluar datos con incertidumbre. El algoritmo *k-PNN* ha sido utilizado ampliamente en diferentes

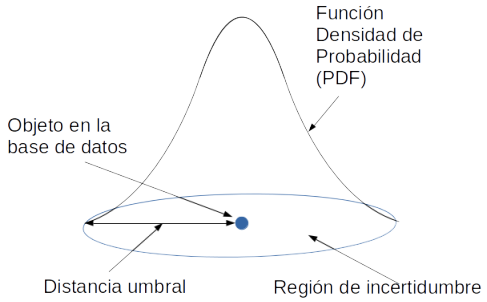



Figura 2.3: Representación de la consideración de la incertidumbre del atributo

aplicaciones, incluyendo servicios de localización [25], monitorización de hábitats naturales [26], análisis de tráfico de red [27], descubrimiento de conocimiento [12] y, ahora por primera vez, en minería de imágenes.

Las consultas con el algoritmo *k-PNN* devuelven una probabilidad no nula, conocida como probabilidad de cualificación, para cada conjunto de $k$ objetos vecinos más cercanos, dado un punto $q$. Dada una base de datos $D$, con $n$ objetos con incertidumbre, donde $D = o_1, o_2, \ldots, o_n$, se puede considerar que el algoritmo *k-PNN* devuelve una lista de respuestas $(s, p(S))$, donde $S$ es un subconjunto de $D$ de cardinalidad $k$, y $p(S)$ es la probabilidad donde los $k$ objetos que forman $S$ son los $k$ vecinos más cercanos de $q$.

La figura 2.4 muestra un ejemplo de *k-PNN*, evaluado sobre 8 elementos con incertidumbre, $(o_1, o_2, \ldots, o_8)$. Si $k$=3, la consulta devuelve un conjunto de tuplas de tres elementos, junto con la probabilidad estimada para satisfacer la consulta. Se ha de tener en cuenta que el número de subconjuntos de $k$ elementos que satisfagan la consulta puede crecer de forma exponencial y puede ser necesario aplicar restricciones adicionales. Por ejemplo, devolver los objetos cuyas probabilidades son más altas que algún umbral con el fin de limitar el tamaño de la respuesta.

El cálculo de *k-PNN* es normalmente más complejo que su homólogo para datos sin incertidumbre. Por ejemplo, calculamos la probabilidad de que $o_1$, $o_2$, $o_5$ sean los tres vecinos más cercanos a $q$ en la Figura 2.4. Como el valor de cada objeto no es exactamente conocido, es necesario considerar la región de incertidumbre. Además, la probabilidad de cualificación de $o_1$, $o_2$ y $o_5$ no depende solamente de los valores de los objetos en sí. También depende de los valores relativos de los demás objetos,

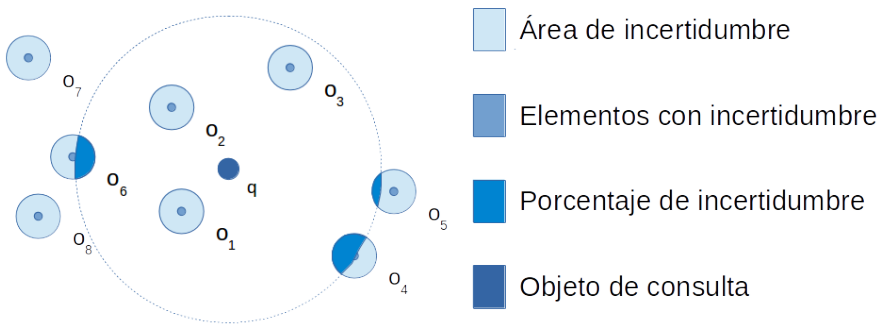



Figura 2.4: Consulta probabilística de $k$-$NN$ ($k$-$PNN$) con $k = 3$.

como por ejemplo $o_3$. El problema se agrava con un mayor número de combinaciones de objetos. Por ejemplo, para evaluar 3-$PNN$ sobre los 8 objetos en la Figura 2.4, se deben calcular las probabilidades para $C_3^8 = 56$ posibles respuestas. El número de respuestas que satisfacen la consulta crece rápidamente. Es obvia la necesidad de una metodología eficiente para gestionar este tipo de consultas.

Supondremos que para la Figura 2.4, por cada instancia se requiere una respuesta con al menos un 20 % de probabilidad, donde los conjuntos $o_1$, $o_2$, $o_3$ y $o_1$, $o_2$, $o_4$ serán las únicas posibilidades, lo que simplifica los cálculos a realizar.

Por ello, en [24] se desarrolla la variante de $k$-$PNN$, con una restricción de umbral de probabilidad $T$, que se conoce como $T$-$k$-$PNN$, de sus siglas en inglés *Probability Threshold k-Nearest-Neighbor Query*. La restricción del umbral $T$, probabilidad de confianza mínima necesaria, permite al usuario controlar el nivel deseado de confianza requerido a una respuesta de una consulta. En la Figura 2.4, con $T$=20 % por ejemplo, un $0,2$-$3$-$PNN$ devuelve $o_1$, $o_2$, $o_3$ y $o_1$, $o_2$, $o_4$ como respuesta a la consulta. Con un valor moderado de $T$, el número de $k$-subconjuntos devueltos es notablemente más pequeño.

### 2.4.4. Métricas de distancia

Los métodos basados en vecindad dependen principalmente de la definición de la distancia entre los elementos. A continuación, se muestran los modelos matemáticos correspondientes a diferentes métricas de distancias, que pueden considerarse para su implementación en el cálculo de $k$ vecinos más cercanos. Para ello, se consideran

dos puntos, $p$ y $q$, en un espacio multidimensional tal que $p = p_1, ..., p_n$ y $q = q_1, .., q_n$ [28]:

- Distancia Kullback-Leibler: $d(p,q) = \sum_{i=1}^{n} p_i \times \log_2 \left(\frac{p_i}{q_1}\right)$
- Distancia Euclídea: $d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$
- Distancia de Manhattan: $d(p,q) = \sum_{i=1}^{n} |p_i - q_i|$
- Distancia de Mahalanobis : $d(p,q) = \sqrt{(p-q)^T V^{-1}(p-q)}$ donde $V$ es la matriz de covarianza
- Distancia de Minkowski : $d(p,q) = (\sum_{i=1}^{n} |p_i - q_i|^r)^{\frac{1}{r}}$ donde $r$ factor de escala

## 2.5. Medidas de evaluación

Como se ha visto anteriormente, la minería de datos dispone de varias tareas que requieren diferentes medidas para su correcta evaluación.

En la clasificación, resulta de gran interés la precisión entendida como el número de instancias clasificadas correctamente dividido por el número de instancias clasificadas como esa clase [29].

En las reglas de asociación se utilizan los conceptos de cobertura: número de instancias a las que la regla se aplica y predice correctamente; y confianza, la proporción de instancias que la regla predice correctamente [30].

En la regresión, se calcula el error cuadrático medio del valor predicho respecto al valor real [31].

Por último, en el clustering, las medidas están relacionas con conceptos de cohesión y distanciamiento entre los grupos [32].

Aunque todas las medidas anteriores son válidas, siempre conviene evaluar el contexto donde se va a utilizar el modelo generado. Cuando se requiere conocer el tipo de error y su coste asociado, se utilizan las matrices de confusión y de coste. Si se considera que todos los errores no son iguales, se puede utilizar el análisis ROC, Receiver Operating Characteristic. Aun utilizando estas nuevas medidas, se debe contrastar el conocimiento previo del problema con el conocimiento generado por el modelo para detectar y resolver posibles problemas.

La matriz de confusión es una herramienta de visualización en la que se representan los resultados obtenidos en una tarea de clasificación. La matriz de confusión es una matriz $nxn$, donde $n$ es el número de clases a representar y se muestra la relación entre la clase perteneciente y la clase asignada. Dicho de otra manera, muestra cuántas instancias se asignan a una clase concreta. En la matriz de confusión, las columnas representan las diferentes clases a las que se puede asignar la instancia y las filas representan la clase asignada de la instancia. De este modo, en una clasificación perfecta, las instancias se situarían únicamente en la diagonal principal, donde cada instancia sería clasificada en su correspondiente clase.

A partir de la matriz de confusión, se derivan las siguientes medidas estadísticas, para evaluar el rendimiento del sistema en términos de recuperación de información: precisión, sensibilidad (en inglés, *Recall* ), F1 y exactitud (en inglés, *Accuracy*).

La precisión mide la probabilidad de que un objeto, clasificado en una clase, realmente pertenezca a esa clase. La sensibilidad, por su parte, mide la probabilidad de que, si un objeto pertenece a una clase, el sistema lo asigne a esa clase. Por su parte, F1 es una relación entre la precisión y la sensibilidad, también conocida como medida armónica. La exactitud se traduce como la proximidad de la medida de los resultados al valor verdadero.

$$Precision = \frac{tp}{tp + fp} \tag{2.1}$$

$$Sensibilidad = \frac{tp}{tp + fn} \tag{2.2}$$

$$F1 = 2\,\frac{Precision \cdot Sensibilidad}{Precision + Sensibilidad} \tag{2.3}$$

$$Exactitud = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.4}$$

$$\tag{2.5}$$

Tal y como se aprecia en las ecuaciones 2.1, 2.2, 2.3 y 2.4, para obtener estos valores estadísticos es necesario el cálculo de ciertos parámetros a través de la matriz de confusión, $tp$, $tn$, $fp$ y $fn$.

Considerando un problema multiclase para una clase dada $tp$, los resultados verdaderos positivos, de sus siglas en inglés *true positive*, son casos en los que a un

objeto se le asigna la clase correcta. Los resultados verdaderos negativos, $tn$ de sus siglas en inglés *true negative*, son el conjunto de objetos asignados correctamente a otras clases diferentes a la clase dada. Los resultados falsos positivos, $fp$ de sus siglas en inglés *false positive*, son el conjunto de objetos pertenecientes a otras clases asignados a la clase dada. Finalmente, los resultados falsos negativos, $fn$ de sus siglas en inglés *false negative*, son el conjunto de datos de una clase dada asignados a otras clases.

# Capítulo 3

# Estado del arte

## 3.1. Estado del arte sobre mapeo temático

El trabajo presentado en esta tesis abarca diferentes ámbitos como son la teledetección y la clasificación supervisada de imágenes a través de una plataforma web, así como su implementación en la nube. En esta sección, se presentan las publicaciones más relevantes encontradas en la literatura existente relacionadas con el trabajo realizado.

En lo referente a la algoritmia para la clasificación de imágenes en el ámbito de la teledetección, las metodologías de análisis se agrupan en sistemas basados en píxeles, basados en objetos e híbridas.

Schröder [33] hace uso de la entrada proporcionada por el usuario en un entorno de aprendizaje bayesiano para la clasificación supervisada de imágenes y la búsqueda de imágenes relevantes.

Costa [34] presenta una clasificación supervisada por píxel junto con un procesamiento post-clasificación con segmentación de imagen y la generalización del mapa semántico. Los resultados muestran que la segmentación de imágenes de alta resolución espacial y la generalización del mapa semántico pueden utilizarse en un contexto operacional para generar mapas de cobertura del suelo automáticamente.

En [35] se evalúa el rendimiento de dos clasificadores, SVM y SAM *spectral angle mapper*, basados en píxeles, para la clasificación de diferentes clases de cobertura de suelo, especialmente en la cobertura vegetal, en entornos urbanos utilizando imágenes

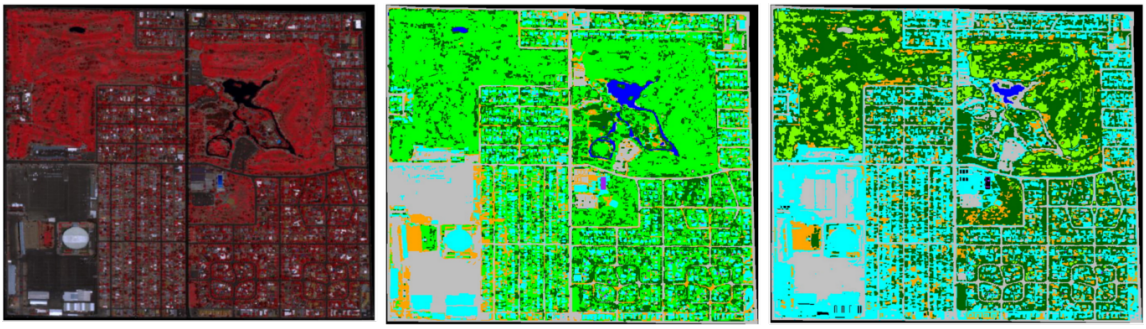


Figura 3.1: Mapa de resultado para clasificación en [36]. De izquierda a derecha, imagen original tomada por el satélite QuickBird. Resultados obtenidos mediante clasificaciones basadas en objetos y en píxeles. Los elementos detectados son edificios(cyan), suelo no administrado (naranja), hierba (verde claro), otras superficies impermeables (gris), piscinas (púrpura), árboles y arbustos (verde oscuro) y lagos y estanques (azul).

hiperespectrales de EO-1 Hyperion.

Myint [36] compara los dos enfoques, basados en objetos y basados en píxeles, con imágenes del sensor QuickBird sobre el área de Arizona. El estudio demuestra que la metodología basada en análisis de objetos mejora significativamente los resultados obtenidos por el análisis basado en píxeles. Este enfoque que utiliza descriptores geométricos, complementando la información radiométrica, es utilizado en el presente trabajo de tesis para caracterizar las imágenes de muy alta resolución.

En [37] se presenta una combinación para la extracción de características basada en objetos, para la búsqueda de edificios en imágenes precedentes y posteriores a desastres y evaluar los daños sufridos. Mediante una clasificación supervisada, evalúa los daños mejorando los resultados obtenidos por inspecciones visuales y otros métodos automáticos.

La combinación de técnicas se puede encontrar en el trabajo desarrollado por Maulik [38]. Los autores proponen un algoritmo de *clustering* paralelo escalable, utilizando la distancia basada en el punto de simetría sobre imágenes multiespectrales de teledetección. La distancia es calculada mediante un algoritmo de búsqueda del vecino más cercano basado en árboles $k$-d.

La interacción visual aplicada a las tecnologías de teledetección es otro aspecto que está adquiriendo gran interés por parte de la comunidad científica. Quan [39]

Figura 3.2: Mapa de resultado para clasificación de daños en catástrofe medioambiental (verde sin daños, amarillo está dañado y rojo destruido) [37].

presenta un *framework* con el que reducir el tiempo y el esfuerzo necesarios en el desarrollo de aplicaciones web para análisis geoespaciales visuales, proporcionando un conjunto de visualizaciones geográficas y representaciones de información.

Keel [40], por su parte, muestra un entorno de Analítica Visual para actividades colaborativas de análisis para la toma de decisiones sobre imágenes de teledetección. El sistema dispone de agentes que infieren relaciones entre la información recogida por los usuarios mediante el análisis de la organización espacial y temporal. En este tipo de trabajos, que requieren la interacción del usuario, el diseño de la interfaz de usuario adquiere gran relevancia.

Por lo que se refiere al uso de sistemas de calculo paralelo para el análisis de datos de teledetección, en [41] se presenta la integración de una cadena para desmezclado de imágenes hiperespectrales con un WCPS, para un entorno de procesamiento de imágenes en la nube, como parte de los servicios web de la suite de la NASA SensorWeb. La implementación en la nube del WCPS permite procesar rápidamente grandes cantidades de datos generando resultados con un bajo coste, como es el caso del presente trabajo.

En [42], se realiza un comparativa de algoritmos de aprendizaje automático para mapeo geológico.

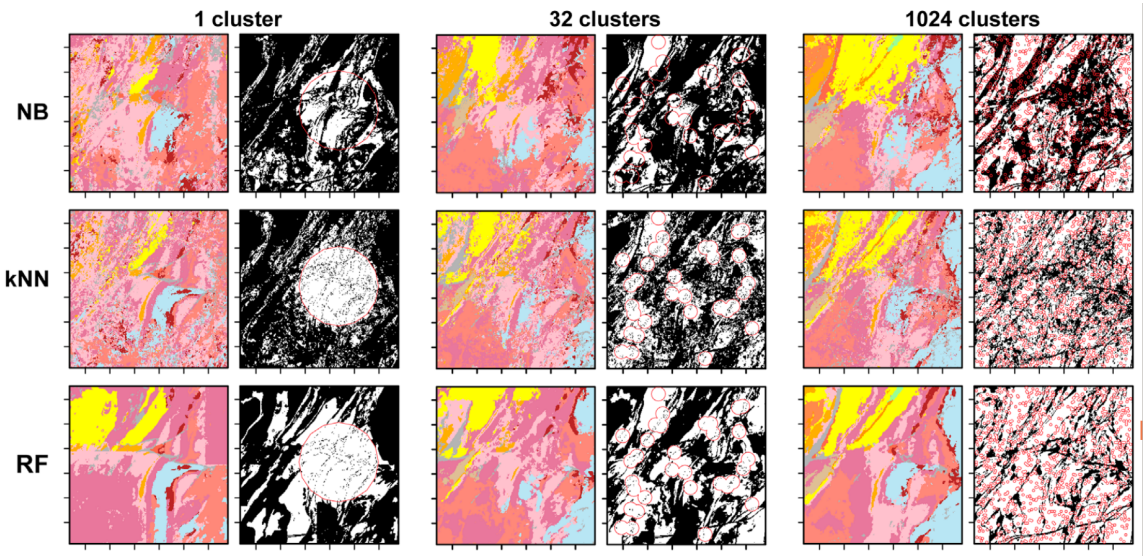



Figura 3.3: Comparativa de resultados de algoritmos para mapeo geológico en [42]

Son múltiples los campos de aplicación de este tipo de desarrollos tal y como se mencionaba en el capítulo anterior. La agricultura, los servicios forestales, la geología, la hidrología y la meteorología son solo algunos ejemplos. A continuación, se presentan diferentes desarrollos relacionados con el presente trabajo según su campo de aplicación.

En el campo de la agricultura, la clasificación de diferentes cultivos y su evaluación requeriría una caracterización especifica de las imágenes. La similaridad visual existente entre cultivos requiere un profundo análisis de la literatura existente, en búsqueda de técnicas que permitan la identificación de patrones o características relevantes. En estas técnicas, la caracterización debe ser lo suficientemente buena para poder discernir entre especies. Este tipo de aplicaciones de teledetección en el campo de la agricultura se pueden encontrar en [43, 44].

En el ámbito forestal, al igual que en el campo de la agricultura, la clasificación de las diferentes especies, el control de la deforestación y el estado de los bosques pueden considerarse como nuevos casos de uso. Otras aplicaciones, como la medición de masas boscosas, requerirían nuevos desarrollos que podrían considerarse tras una clasificación previa. Ejemplos sobre la clasificación de bosque los encontramos en [45–49].

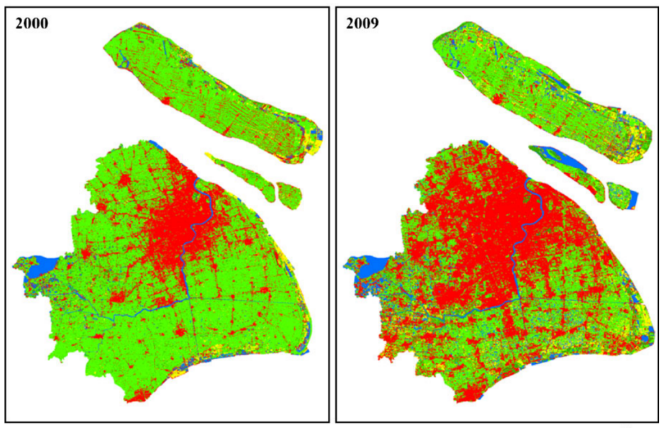



Figura 3.4: Uso y cobertura de la superficie durante en 2000 y 2009 en Shangai de [53]. Amarillo para suelo desnudo, verde claro para cultivos, azul para masas de agua, verde oscuro para zonas boscosas y rojo para áreas urbanas

## 3.2. Aplicaciones de mapeo temático

El control medioambiental es otro dominio de aplicación claro. A las ya mencionadas clasificación de especies y control del estado de las masas forestales, podrían añadirse aplicaciones para el control de costas y protección de cuencas de ríos. El control de costas es una tarea que diferentes autoridades llevan a cabo para el mantenimiento de hábitats naturales protegidos o control sobre actividades humanas cada vez más agresivas como la urbanización de las mismas. La clasificación del uso del terreno, puede ser un caso de uso relativamente sencillo de aplicar. La monitorización, tanto de costas, como el uso de la tierra, es una aplicación típica en la teledetección como se puede ver en los trabajos presentados en [50–54] y la clasificación de la mismas en [55].

Haciendo uso, ya no solo de imágenes ópticas, sino de imágenes hiperespectrales, se puede extraer información sobre la composición del suelo y el subsuelo basándose en reflectancia espectral para su clasificación, mapeo litológico o clasificación del suelo, como muestran los siguientes trabajos [56–59].

La teledetección a través de imagen radar ofrece una mejor visión sobre la distribución y la dinámica de los fenómenos hidrológicos, impensables en estudios tradicionales sobre el terreno. Posibles aplicaciones como mapeo de humedales e inundaciones no requieren grandes cambios, mientras que aplicaciones que requieran el uso

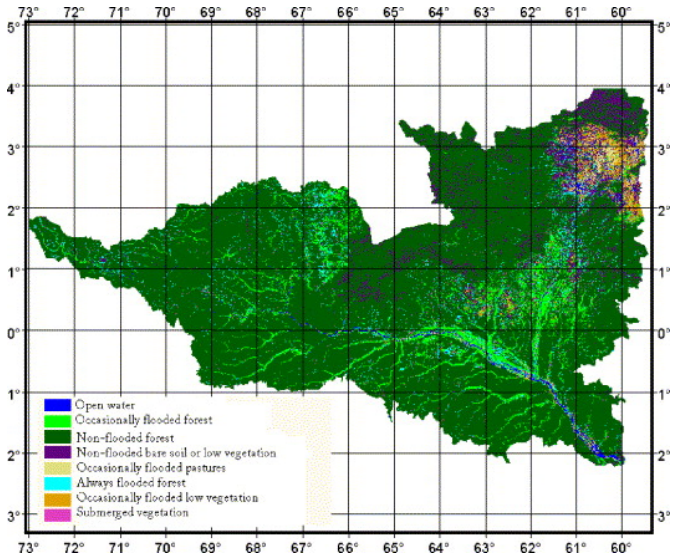



Figura 3.5: Resultados de clasificación de la cuenca del Rio Negro basados en imágenes radar de JERS-1 [64]

de imágenes radar e hiperspectrales, como la medición del espesor de la nieve, requerirán nuevos desarrollos. Parece que una tarea relacionada con este ámbito como es el seguimiento de grandes masas de hielo -muy demandada por múltiples organismos e industrias- apenas requeriría de excesivos cambios. Podemos encontrar diferentes ejemplos de aplicaciones de uso con carácter hidrológico en los siguientes trabajos [60–65].

Tal y como se comenta en la sección 2.2, el tiempo de procesamiento, entendiéndolo como la suma de los tiempos de entrenamiento y test del modelo algorítmico de carácter adaptativo, es de suma importancia. El estudio realizado en [42] realiza una comparativa entre cinco algoritmos de clasificación para el mapeo geológico utilizando imágenes de teledetección, incluyendo el tiempo de procesamiento, Figura 3.6. Los algoritmos evaluados son Naive bayes, $k$- Vecinos más cercanos, *Randon Forest*, *SVM* y una red neuronal del tipo Perceptrón Multicapa. Obteniendo unos valores de exactitud similares, el tiempo de procesamiento del algoritmo $k$-NN se sitúa en torno al minuto, mientras que el tiempo de procesamiento va desde los tres a los 25 minutos en los demás algoritmos. Ésta es una de las razones para la selección de este método en el trabajo presentado en esta tesis.

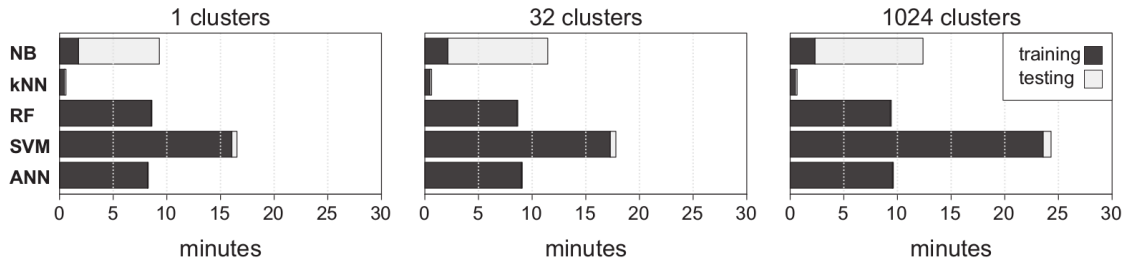



Figura 3.6: Comparativa de tiempo de procesamiento entre algoritmos para el mapeo geológico [42].

## 3.3. Estado del arte sobre mapeo temático a través de entornos web

A continuación se citan algunas aplicaciones sobre teledetección con interfaz de acceso web. En [66] se presenta el caso de uso para la detección temprana de inundaciones mediante la plataforma *SensorWeb*, haciendo uso de los datos recogidos sobre Namibia por el sensor Earth Observation One (EO-1). Compuesta por diferentes tipos de sensores espaciales, aéreos y terrestres, proporciona un acceso fácil a los datos y resultados automatizados rápidamente, permitiendo realizar predicciones sobre inundaciones y actuar en consecuencia.

En [67], se describe ncWMS, una implementación de interoperable con los estándares para los WMS descritos por el OGC para la visualización e interacción de información medioambiental multidimensional. Diseñada para funcionar con una configuración mínima, no requiere la descarga de grandes cantidades de datos ni la interpretación de datos complejos y hace de puente entre la comunidad medioambiental y usuarios de herramientas GIS.

En [68], se presenta la aplicación web OWGIS para la creación de sitios web mediante código HTML, JavaScript y archivos XML donde se definen las capas de datos geográficos. Siguiendo los estándares OGC, es capaz de solicitar datos a servidores como GeoServer o ncWMS, permitiendo analizar, visualizar, compartir o comparar datos. El perfil de científico medioambiental es el más común entre los usuarios de la aplicación.

[69] realiza un estado del arte sobre los recursos necesarios para la realización de aplicaciones sobre recursos hidrológicos vía web. Se hace especial hincapié en los

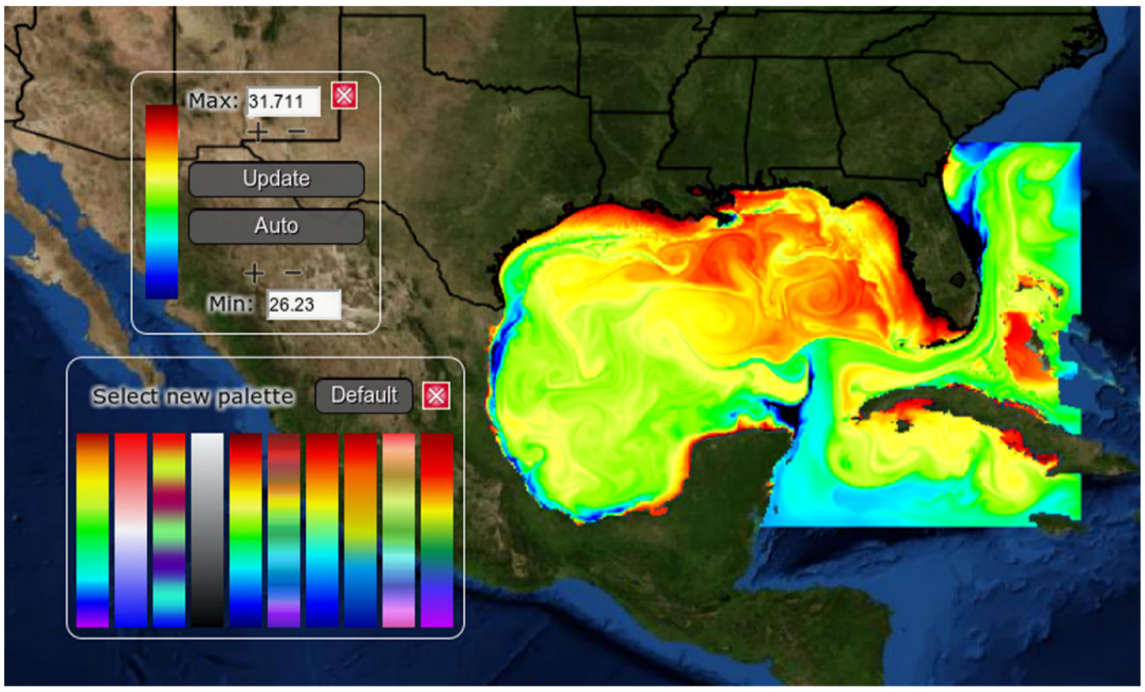



Figura 3.7: Visualizador de datos GIS vía web [68].

requerimientos del análisis de datos espaciales y en las herramientas de software libre disponibles para ello, desde bases de datos, librerías de mapeo, lenguajes de programación etc.

En [70], se describe y evalúa un sistema automatizado de dos fases para el control de inundaciones, desarrollado por DLR o la agencia espacial alemana, como soporte de la gestión rápida de desastres. En la primera fase se detectan inundaciones potenciales mediante el análisis de datos ópticos del sensor MODIS mientras que la segunda fase se activa mediante el análisis de imágenes SAR. El sistema realiza un tratamiento previo de datos, cálculo y la adaptación de los datos auxiliares, clasificación temática y difusión de mapas de inundaciones mediante un cliente web interactivo. La operabilidad del sistema está demostrada y evaluada mediante la monitorización de dos inundaciones recientes, en Rusia durante 2013 y en Albania/Montenegro, en 2013.

[71] presenta un sistema de mapeo web que permite a usuarios no expertos realizar clasificaciones no supervisadas sobre imágenes no multiespectrales de teledetección. El sistema de procesamiento síncrono está basado en la clasificación no supervisada mediante los algoritmos ISODATA y kmeans. Basado en un servidor web Apache,

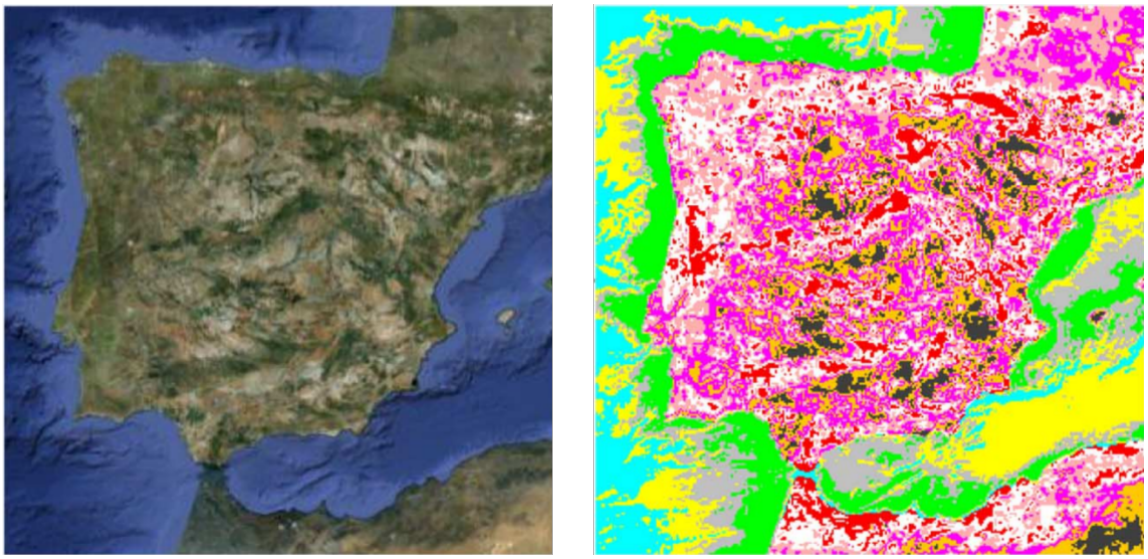


Figura 3.8: Ejemplo de resultado sobre la península ibérica de [71].

está implementado mediante tecnologías HTML5, JavaScript, Php y AJAX. La aplicación online hace uso de la API de Google Maps, lo que facilita la navegación entre diferentes niveles de zoom.

## 3.4. Procesamiento y análisis de grandes volúmenes de imágenes de teledetección

Debido al continuo crecimiento del volumen de datos en teledetección y el número de usuarios de estos, se requieren mecanismos que permitan adquirirlos, distribuirlos y procesarlos eficientemente, por lo que el desarrollo de estos mecanismos resulta crítico [72]. Para hacer frente a estas necesidades, recientes investigaciones se han centrado en técnicas de procesamiento de alto rendimiento en el ámbito de la teledetección [3, 73].

Estas técnicas de alto rendimiento, que integran entornos y técnicas de programación, facilitan la resolución de problemas a gran escala, como los encontrados en la teledetección, son cada vez más utilizados.

En este sentido, para conseguir una respuesta eficaz a las consultas, la organización de los datos es fundamental. En particular, la búsqueda del vecino más cercano

puede beneficiarse de las estructuras jerárquicas de indexación [74]. Los árboles $k$-d son estructuras de datos que particionan el espacio para la organización de los elementos en espacios euclídeos $k$-dimensionales. Se basan en conjuntos de hiperplanos perpendiculares a cada uno de los ejes del sistema de coordenadas.

En [3] se hace un repaso de los diferentes enfoques que la comunidad científica ha venido implementando en los últimos años mediante: procesamiento en hardware especializado, procesamiento en clusters o procesamiento en infraestructuras distribuidas [75].

Los equipos que forman parte de un sistema de computación distribuida pueden ejecutar diferentes sistemas operativos y tener un hardware diferente mientras que, en los clusters, todos los equipos tienen el mismo hardware y sistema operativo. La computación distribuida puede llegar a utilizar las capacidades de procesamiento de un equipo de escritorio, mientras que los equipos de un cluster trabajan como un solo equipo. La computación distribuida, por su naturaleza, se distribuye a través de una LAN o WAN mientras que, en un cluster, los equipos normalmente se encuentran en el mismo lugar o instalación. Por sencillez, el trabajo descrito en esta tesis se ha enfocado exclusivamente desde el punto de vista del uso de clusters para el mapeo temático.

Una de las mayores ventajas de estos sistemas está en la separación entre la capa lógica y funcional, y la capa física. Esta separación libera al usuario de gestionar recursos e infraestructuras, permitiéndole centrarse en el desarrollo y análisis de su trabajo. Otro aspecto importante es la escalabilidad de estos sistemas, tanto en respuesta a cambios de requerimientos de los sistemas, como frente a demandas de aumento de recursos. La evaluación de esta escalabilidad, para el caso concreto de un sistema de mapeo temático, se considera en el Capítulo 6 del presente trabajo.

El paradigma de programación paralela *MapReduce* se basa en un sistema de ejecución para el reparto de datos de entrada de forma automática y la distribución de los resultados intermedios [76] sobre los nodos de un cluster. Técnicamente, *MapReduce* ejecuta un *Mapper* para procesar los datos de entrada de forma paralela y producir unos resultados intermedios construidos en base a una serie de pares clave/valor, mientras que el *Reducer* combina los valores de los resultados intermedios asociados con la misma clave.

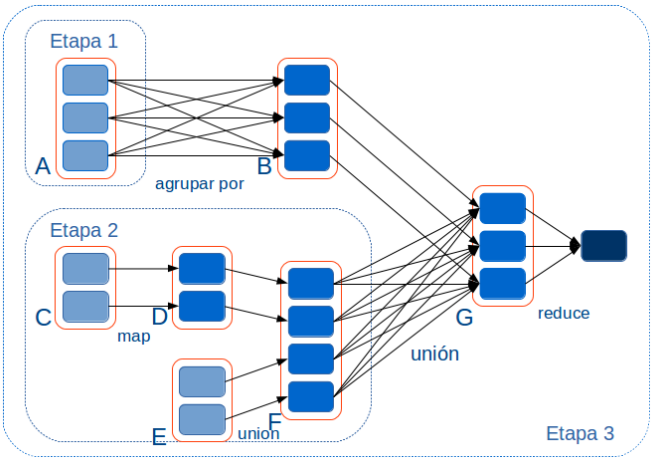



Figura 3.9: El diagrama muestra cómo se secuencian operaciones complejas en Spark [78]. Cada rectángulo naranja es un conjunto de datos distribuido (RDD) y los rectángulos internos representan las particiones. Los RDD se relacionan con otros mediante transformaciones. Las más sencillas, como la operación *map*, ejecutan la operación en cada partición independientemente y en paralelo. Otras transformaciones, como la agrupación, requieren mover los datos por las particiones. Los rectángulos claros cargan los datos directamente desde el disco, mientras que los oscuros son objetos intermedios que pueden cachear datos en memoria. Cuando un usuario solicita la salida del RDD final, el más oscuro de los rectángulos azules, el gráfico de operaciones, se compila en tres etapas.

La versión *MapReduce open source* desarrollada por Yahoo! en 2006, conocida como Hadoop [77], hizo accesible este modelo de programación al público en general y fue ampliamente adoptado por la industria. Pero la implementación *MapReduce* de Hadoop tiene sus limitaciones. Por un lado, los datos son cargados desde el disco para cada análisis, lo que puede resultar demasiado lento si se quieren implementar procesos que requieran repetidos accesos a los datos, como las operaciones iterativas típicas de los algoritmos de aprendizaje automático. Por otro lado, la concatenación de diferentes operaciones, en forma de diferentes flujos de trabajo, puede resultar muy costosa, como por ejemplo, el registro de imágenes en tiempo real, cálculos estadísticos de cada imagen, análisis de imágenes, etc., etc. Además de ser complejas de expresar, pueden terminar en implementaciones ineficientes.

La plataforma *Spark*, utilizada en el presente trabajo como alternativa a Hadoop, ha sido desarrollada en el departamento AMPLab de la universidad de Berkley en

2009 [78]. *Spark* soluciona varias de las limitaciones comentadas anteriormente mediante una nueva abstracción conocida como *resilient distributed dataset*, de sus siglas en inglés *RDD*, y su correspondiente motor de ejecución. *RDD* es una colección de registros distribuidos (palabras, cuadrículas de imágenes, etc.), que pueden ser procesados en paralelo mediante operadores de alto nivel. Cuando un usuario encadena una secuencia de operaciones, la implementación subyacente compila el grafo de las operaciones encadenadas en una serie de pequeñas y eficientes tareas como se aprecia en la Figura 3.9.

Desde la perspectiva del usuario, facilita la especificación de la localización de los datos a cargar y las operaciones a realizar, mientras que Spark gestiona tareas para la ejecución de la secuencia de ejecución a lo largo del cluster (Figura 3.10). *Spark* permite *cachear* los datos y almacenarlos momentáneamente en la RAM distribuida por el cluster, permitiendo repetidas consultas rápidas. Esta característica es especialmente importante debido a que la carga y recarga de los datos suele ser, por lo general, el cuello de botella en sistemas basados en algoritmos iterativos y recursivos como los que caracterizan el aprendizaje automático. De este modo, las imágenes *raw*, o imágenes en bruto, pueden ser cargadas y cacheadas para una posterior secuencia compleja de operaciones o repetidos análisis interactivos, sin la necesidad de recargarlas desde el disco. Finalmente, la API de *Spark*, para desarrollos en Java, Scala y Python, permite expresar operaciones complejas intuitivamente, con una mínima cantidad de código.

Trasladar el aprendizaje automático al ámbito del *cloud computing* requiere trasladar los algoritmos de aprendizaje automático y adaptarlos con el fin de explotar las capacidades de ejecución distribuida sobre grandes volúmenes de datos. Mientras que la paralelización y la distribución de los procesos de análisis proporcionan ventajas evidentes, trasladar los clásicos algoritmos de aprendizaje automático destinados a volúmenes de datos limitados a entornos de teledetección con datos de cobertura a gran escala supone un desafío. La adopción de enfoques de paralelización basados en el paradigma *MapReduce* implementado bajo *Spark* se plantea como una solución al problema descrito.

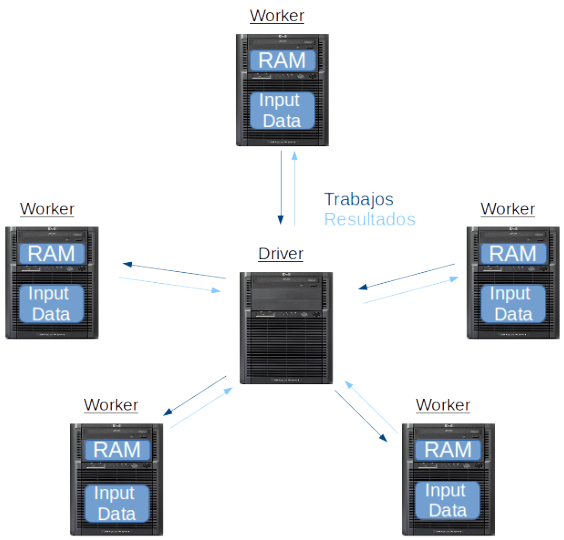



Figura 3.10: Modo de ejecución de Spark [78]. El programa principal lanza múltiples *workers*, que leen bloques de datos desde un sistema de archivos distribuido y pueden mantener particiones RDD en memoria.

## 3.5. Caracterización de los datos

La caracterización de los datos está basada en descriptores de imágenes, ampliamente desarrollados a lo largo de la literatura sobre sistemas de recuperación de imágenes basadas en contenido *CBIR*, de sus siglas en ingles *Content-based image retrieval*, donde se dedican importantes esfuerzos a una cuidadosa elección y aplicación de descriptores de contenido de la imagen [79, 80]. Los descriptores basados en características primitivas abarcan una amplia gama, desde descriptores a nivel de píxel como el color, a descriptores geométricos como la textura [81–83], siendo habitual combinaciones de las mismas [1, 84].

Los descriptores globales o a nivel de imagen son habitualmente complementados con descriptores locales a nivel de regiones. Mientras que los primeros tienen propiedades deseables para la discriminación semántica a nivel de escena, los últimos posibilitan la caracterización y reconocimiento de elementos específicos dentro de la escena. La composición adecuada de estrategias discriminatorias en un contexto semántico es objeto de un gran número de investigaciones [1, 85, 86].

Los descriptores de color son de las características más utilizadas en los siste-

mas *CBIR* [87]. Su utilización requiere la selección de un espacio de color entre las diferentes opciones existentes: RGB, HSV, HIV, CIE, CMYK, YUV, YCbCr, etc [88].

La textura se considera como la repetición periódica o cuasi-periódica de un patrón local básico en un cierto área [89]. En la literatura nos podemos encontrar con diferentes enfoques para el análisis de texturas: mediante métodos estadísticos, métodos geométricos, basados en modelos o basados en tratamiento de señal. A continuación, se consideran los descriptores utilizados a lo largo del presente trabajo de tesis doctoral.

El descriptor de Histograma de Gradientes Orientados *HOG* [90], de las siglas en inglés *Histogram of Oriented Gradients*, es uno de los descriptores usados en teledetección. Se basa en que la apariencia y la forma de un objeto pueden ser descritos por la distribución de las direcciones de los gradientes. La imagen original se divide en bloques, de los que se obtienen los histogramas de gradientes orientados. Para ello, la imagen se transforma a escala de grises y se divide en bloques. Por cada píxel del bloque, se obtiene el gradiente correspondiente, compuesto por magnitud y orientación. Se crea un histograma basado en las orientaciones, en la que por cada orientación, se acumula la magnitud, aplicando un peso de los píxeles. A continuación, se normaliza el histograma. El descriptor de la imagen está compuesto por los histogramas de los bloques. De este modo, si se consideran 8 orientaciones y la imagen se divide en 10 bloques, el descriptor sería un vector de $8 \times 10$.

El descriptor *LBP* [96] consiste originalmente en la suma de la comparación del píxel central, de una ventana $3 \times 3$, con los vecinos. En caso de que el vecino tenga un valor mayor, se acumula un valor del peso asignado a ese vecino. El histograma generado a partir de cada uno de los valores de los píxeles puede utilizarse como descriptor de textura.

El acrónimo *SIFT*, de las siglas en inglés *Scale-Invariant Feature Transform*, extrae las características relevantes de la imágenes para el reconocimiento de objetos [97]. El descriptor *SIFT Density* representa la densidad de puntos relevantes de una imagen de forma que describe una textura.

El descriptor de textura *Edge Density* [98] [99], o densidad de bordes, se basa en la búsqueda de transiciones bruscas de los niveles de gris, que representan los bordes.

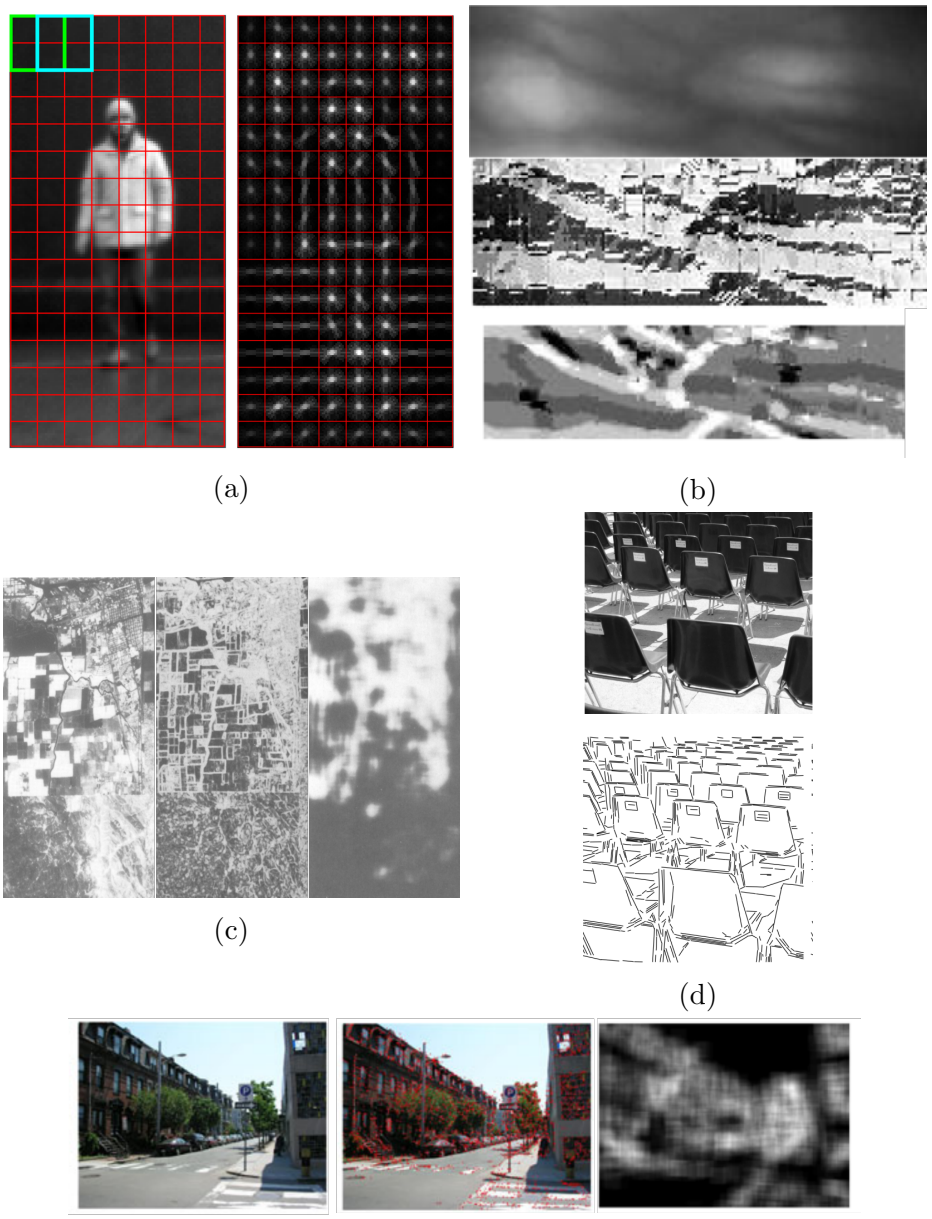


Figura 3.11: Ejemplos sobre imágenes genéricas de descriptores de textura utilizados en teledetección [80]. (a) descriptor de textura *HOG*, (b) descriptor de textura *LBP*, (c) descriptor de textura *Edge Density*, (d) descriptor de textura *LSD*, (e) descriptor de textura *SIFT Density*, ejemplos aplicación de [91–95]

Para una distancia $d$, se calcula la densidad de bordes detectados.

Los descriptores de ángulos o segmentos de línea *LSD* [100], *Right-Angle Detector* o *Line Segment Detector*, también están basados en detecciones de variaciones bruscas en los niveles de gris, de las que se obtiene el número de segmentos de línea detectados.

## 3.6. Mapas de validación sobre imágenes hiperespectrales de teledetección

La evaluación de los resultados de un sistema de mapeo temático se realiza mediante la comparación de los resultados obtenidos con mapas de validación preexistentes. En el ámbito de la teledetección existen mapas de validación de referencia ampliamente utilizados por la comunidad científica a lo largo de diferentes trabajos.

Típicamente, la evaluación de los sistemas se realiza utilizando un subconjunto de píxeles o regiones del total de los mapas generados a partir del cual se obtienen diferentes medidas estadísticas que permiten evaluar el rendimiento del sistema, tal y como muestra la Figura 3.12 [101].

A continuación, se presentan diferentes conjuntos de datos para la evaluación de los resultados de clasificación sobre imágenes de teledetección.

- Indian Pines, Indiana. La escena captada por el sensor *AVIRIS* sobre el noroeste del estado de Indiana está formada por imágenes de $145 \times 145$ píxeles y 224 bandas espectrales, con una longitud de onda comprendida en el rango de $0,4$ a $2,5 \mu ms$. Dos terceras partes de las imágenes corresponden a terrenos agrícolas y una tercera parte a bosques u otros tipos de vegetación. También se visualizan dos carreteras de doble carril, una línea de ferrocarril, así como algunas viviendas, otras estructuras construidas y carreteras más pequeñas. En la escena se aprecian diferentes tipos de cultivos como maíz y soja. El mapa de validación dispone de 16 casos diferentes. Los datos están disponibles a través del sitio web dedicado a imágenes multiespectrales de la universidad de Purdue [1].

[1] https://engineering.purdue.edu/biehl/MultiSpec/hyperspectral.html

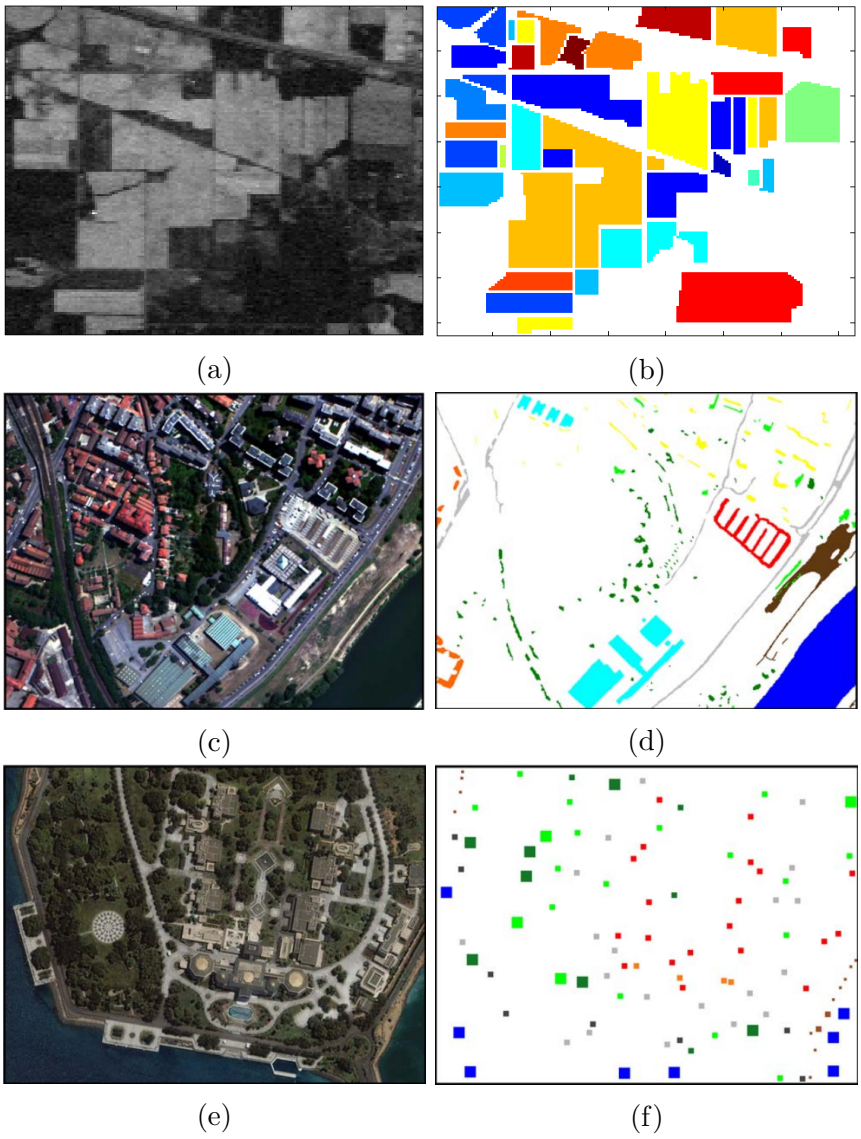



Figura 3.12: Validación para clasificación de imágenes de teledetección. Submuestreo de regiones previamente clasificadas sobre las que obtener medidas estadísticas que evalúen el funcionamiento del aprendizaje automático. Indian Pines (3.12a y 3.12b), Pavia (3.12c y 3.12d) y Jeddah (3.12e y 3.12f) [101]. En todos los casos considerados, la resolución geométrica considerada resulta diferente respecto a la del presente trabajo.

- Universidad y centro de Pavia, Italia. La escena recogida por el sensor *ROSIS* dispone de 103 bandas hiperespectrales para un tamaño de imagen de $601 \times 610$ píxeles con un resolución de 1,3m por píxel en la que se distinguen hasta 9 clases. Los datos están disponibles a través de la web del Grupo de Inteligencia Computacional de la Universidad del País Vasco (UPV/EHU) [2].

- Jeddah. Este conjunto de datos esta compuesto por imagenes multiespectral de muy alta resolución adquirida por el sensor IKONOS en julio de 2004. La imagen tiene tres bandas espectrales con una resolución espacial de 1m, y se refiere a una parte de la ciudad de Jeddah (Arabia Saudí), en el que ocho tipos de cobertura del suelo son dominantes: dos tipos de asfalto, suelo desnudo, hierba, dos tipos de techos, árboles y agua.

En la Figura 3.12 se presentan diferentes ejemplos de los mapas de validación presentados. Cabe destacar la baja resolución de las imágenes y el nivel de detalle de las imágenes de validación, donde una gran parte de la misma no esta asignada a ninguna clase.

Aunque todos estos ejemplos corresponden a mapas realizados para procesamientos basados en imágenes hiperespectrales, sus correspondientes mapas de validación son de utilidad. Lamentablemente, el nivel de resolución de estos mapas no corresponde al nivel de resolución de las imágenes presentadas en la sección 6.1. El inconveniente de la resolución radica en que la máxima resolución que encontramos se refiere al conjunto de mapas de Jeddah, con 1 metro por píxel. Esta resolución es 4 veces menor que los 25 cm por píxel de las imágenes utilizadas en el presente trabajo, lo que repercute directamente en el tipo de elementos que podemos diferenciar.

Por otro lado, tampoco se han encontrado mapas de validación para imágenes ópticas. Esto dificulta la validación de la implementación, teniendo que desarrollar algún sistema que nos permita realizar la evaluación y así poder evaluar el sistema al completo.

[2] http://www.ehu.eus/ccwintco/index.php

## 3.7. Conclusiones del estudio sobre el estado del arte

El desarrollo del presente trabajo de tesis doctoral viene motivado por la necesidad de superar diferentes aspectos mencionados a lo largo del análisis del estado del arte realizado en este capítulo.

Partiendo de métodos de probada eficacia de clasificación en el ámbito de la teledetección, como el $k$ vecinos más cercanos optimizado mediante árboles $k$-d, se ha realizado una caracterización basada en descriptores geométricos para comprobar la viabilidad de la integración de funcionalidades de aprendizaje automático en servidores web y mediante su posterior implementación en la nube en plataformas de clusteres.

El desarrollo requerido y los resultados obtenidos se describen a lo largo de los próximos capítulos.

# Capítulo 4

# Avance sobre el estado del arte

Mediante el desarrollo de un prototipo pre-operacional, se pretende comprobar la viabilidad y los aspectos principales de integración de funcionalidades de aprendizaje automático en servidores web para el mapeo temático sobre imágenes de teledetección.

El modelo aplicativo se ha desarrollado dentro del ámbito de la teledetección, más concretamente en el mapeo temático de imágenes de muy alta resolución. Esto requiere habilitar procesos de supervisión capaces de modelar la incertidumbre derivada de la supervisión mediante procesos probabilísticos a través de interfaces web.

El mapeo temático realizado mediante un esquema de entrenamiento interactivo permite al usuario definir ejemplos que afectan directamente al modelo probabilístico creado para la clase temática de interés, generando un mapa temático personal. Al mismo tiempo, el mapeo temático supervisado implica incertidumbres en forma de errores en los datos e incertidumbres en el entrenamiento proporcionado por el usuario. La gestión de estas incertidumbres requiere algoritmos de clasificación probabilísticos, mientras que la eficiencia operacional requiere que este tipo de algoritmos sean implementados sobre las estructuras de datos eficientes en grandes volúmenes de datos $N$-dimensionales.

En la sección 2.4.3, se presenta el algoritmo implementado, junto con su optimización basada en consultas en bloque. En la sección 3.5, se presenta la caracterización implementada para imágenes de teledetección.

## 4.1. *T-k-PNN para imágenes*

Como se comenta en la sección 2.4.3, el algoritmo *T-P-kNN* ha sido desarrollado para efectuar búsquedas en bases de datos con incertidumbre [102]. Además, el *T-k-PNN* ha sido utilizado en otros entornos, como por ejemplo para modelar la incertidumbre implícita a la localización en interiores para el posicionamiento de objetos en movimiento [103].

Una aportación fundamental del presente trabajo es la verificación de su aplicabilidad en tareas de clasificación supervisada de imágenes: se propone *T-k-PNN* como algoritmo, el cual devuelve conjuntos de $k$ objetos que satisfacen una consulta con una probabilidad más alta que un umbral $T$ para el mapeo temático en teledetección.

Un problema de implementación en este ámbito es que la evaluación de una consulta mediante el algoritmo puede ser computacionalmente costosa mientras se requiera un número creciente de $k$ subconjuntos. Como posible solución, puede considerarse no requerir el valor exacto de la probabilidad y conformarse con un grado de confianza.

El algoritmo de clasificación implementado está estrechamente relacionado con el algoritmo *T-k-PNN*, de las siglas en ingles *Probabilistic Threshold k-Nearest Neighbor*, diseñado para devolver el conjunto de puntos $S$ más probable de la clase $D$ dado un punto $o_i$, de forma que:

$$S|S \subseteq D \bigwedge |S| = k \text{ y } p(S) \geq T, \text{ donde } T \in [0,1].$$

Así, la cualificación de la probabilidad $p(S)$ de un subconjunto $S$ con $k$ elementos se calcula de la siguiente manera [24]:

$$p(S) = \sum_{o_i \in S} \int_0^{+\infty} d_i(r) \prod_{o_j \in S - \{o_i\}} D_j(r) \prod_{o_h \in D - S} (1 - D_h(r))\, dr \tag{4.1}$$

donde la distancia de la *PDF* dado un píxel de entrenamiento $o_i$ se denota como $d_i(r)$, mientras que su Función Densidad Acumulada, o *CDF* de sus siglas en inglés (Cumulative Density Function), se denota como $D_i(r)$, siendo $r \in \Re$ el valor absoluto a la distancia $r_i = |o_i - q|$ a punto $q$ de consulta, y donde se estima la *PDF* mediante estimaciones basadas en núcleo y numéricamente integradas para la *CDF*. La Tabla 4.1 resume los símbolos en la Ecuación 4.1.

| Símbolo | Significado |
|---|---|
| $S$ | Clase objetivo |
| $D$ | Base de datos con incertidumbre, o regiones de imágenes a clasificar |
| $k$ | Número de puntos |
| $p(S)$ | Cuantificación de probabilidad of $S$ |
| $T$ | Umbral de probabilidad |
| $o_i$ | Objeto con incertidumbre $i$ de $D(i = 1, ..., \|D\|)$, regiones de entrenamiento |
| $q$ | Objeto de consulta, región de imagen con clase desconocida |
| $r_i$ | $\|o_i - q\|$, distancia entre el objeto de consulta y el objeto con incertidumbre |
| $d_i(r)$ | $PDF$ de $r_i$ (Función Densidad de Probabilidad de las distancias $r_i$) |
| $D_i(r)$ | $CDF$ de $r_i$ (Función de distribución acumulada de distancias $r_i$), con $r_i$ la distancia relativa a otro objeto de la clase. |
| $D_h(r)$ | $CDF$ de $r_h$ (Función de distribución acumulada de distancias $r_h$), con $r_h$ la distancia relativa hacia otras clases |

Tabla 4.1: Símbolos para la Ecuación 4.1 describen la probabilidad en el algoritmo de clasificación supervisada $k$-Vecinos más cercanos.

Si se considera una consulta múltiple, mediante la ecuación 4.1, ésta requerirá un proceso de unificación entre las diferentes soluciones. La unificación puede llevarse a cabo estimando y minimizando la distancia por píxel $r_i(o_i, q)$ al elemento de entrenamiento más cercano, ya sea en el espacio de características o geográfico. De este modo, obtendremos como resultado una clasificación multiclase relacionada con la dimensión espacial que comparte caracteristicas con una segmentación, por el hecho de que la medida de distancia considerada mezcla el espacio geográfico y el espacio de descriptores.

Los autores de [24] observan que la ecuación mencionada puede entenderse, considerando $S$ como respuesta a la consulta, como la distancia a cualquier objeto $o_h \notin S$ desde $q$ debe ser mayor que la de $o_i$ donde $o_i \in S$. A una distancia $r$, la $PDF$ que el objeto $o_i \in S$ tiene la $k$-esima mínima distancia desde $q$ es el producto de varios factores:

- la $PDF$ de $o_i$ tiene una distancia de $r$ desde $q$, p.e. $d_i(r)$;
- la probabilidad de que todos los objetos en $S$, aparte de $o_i$, tengan menores distancias que $r$, por ejemplo $\prod_{o_j \in S \bigwedge o_j \neq o_i} D_j(r)$
- y la probabilidad de que los objetos en $D - S$, tengan mayor distancia que $r$, p.e. $\prod_{o_h \in D-S}(1 - D_h(r))$.

La función de integración en la Ecuación 4.1, es esencialmente el producto de estos tres factores. Mediante la integración de esta función entre $(0, +\infty)$, obtenemos la probabilidad de que $S$ contenga los $k$ vecinos más cercanos con $o_i$ como el $k$-esimo vecino más cercano. Por último, mediante la suma del valor de probabilidad para todos los objetos $o_i \in S$, se obtiene la Ecuación 4.1.

Los autores observan en su contribución [24] que la Ecuación 4.1 es ineficiente de evaluar, ya que requiere el cálculo de las distancias $PDF$ y $CDF$ en cada objeto mediante una costosa integración numérica para un amplio rango de valores.

La explotación de estructuras de datos eficientes, como los árboles $k$-d, permite al sistema desarrollado una mejora en su rendimiento hasta el punto de soportar consultas a través de una red de datos.

En los árboles $k$-d generados, cada hiperplano se representa mediante un nodo. También se puede considerar que cada nodo representa un subconjunto de elemen-

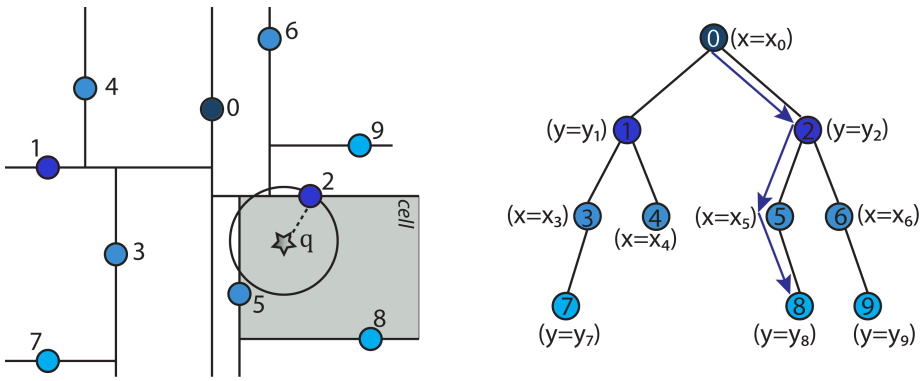



Figura 4.1: Árbol $k$-d y estructura en aŕbol correspondiente

tos. Para cada nodo se elige el atributo más discriminante, de donde se obtiene la mediana de todos los elementos y, a partir de la cual, se divide el conjunto de datos en dos subconjuntos. Un primer conjunto con el valor del atributo menor o igual al de la mediana y un segundo conjunto cuyo valor del atributo es mayor que la mediana. Recursivamente, se crean los árboles binarios para cada nodo, hasta obtener conjuntos con un número mínimo de elementos, previamente definido, desde el nodo principal o raíz, hasta los últimos nodos u hojas. De este modo, los nodos del árbol guardan un punto y un hiperplano divisor del espacio.

Para encontrar de manera eficiente los vecinos más próximos, es necesario definir un ámbito de búsqueda local, que se lleva a cabo por el árbol $k$-d. El proceso de búsqueda recursivo se basa en la comparación del valor de corte de un punto $v$ con el valor correspondiente de una muestra $x$ dada que denominaremos $c$. Siendo $d_{nn}$ la distancia al vecino más cercano:

$$x[c] + d_{nn} \leq v \text{ el nodo hijo izquierdo contendrá el vecino más cercano} \tag{4.2}$$

$$x[c] + d_{nn} \geq v \text{ el nodo hijo derecho contenedrá el vecino más cercano} \tag{4.3}$$

De forma recursiva, se van atravesando los diferentes nodos, hasta llegar al nodo hoja en el que la muestra se compara con todos los elementos del mismo. En la Tabla 4.2, se resumen los símbolos en la ecuación 4.1.

El beneficio clave es la reducción en el coste computacional que supone encon-

| Símbolo | Significado |
|---|---|
| $x$ | Muestra desconocida |
| $v$ | Valor de corte, mediana de las coordenadas discriminantes |
| $c$ | Coordenada de $x$ discriminante para ese nodo |
| $d_{nn}$ | Distancia al vecino más cercano |

Tabla 4.2: Descripción de símbolos para búsqueda optimizada de vecinos más cercanos con árbol $k$-d.

trar el vecino más cercano, pasando de $O(n)$ a $O(log(n))$ de promedio. Esto mejora significativamente el rendimiento cuando se trata de grandes archivos de datos. El algoritmo de construcción del árbol utilizado se describe en [104].

En el proceso de creación de una capa temática, el usuario selecciona diferentes regiones de entrenamiento. Como hemos visto, el resultado de esta selección se modela como una combinación de variables aleatorias en un espacio de características, con una $PDF$ asociada.

Esta aproximación utiliza diferentes parametrizaciones para cada conjunto de entrenamiento. Los resultados finales se han obtenido operando en los árboles $k$-d generados.

En el actual flujo de ejecución del algoritmo optimizado, la ecuación 4.1 requiere ser calculada repetidamente para todos los píxeles de las regiones a clasificar. Se utiliza un mecanismo de cacheo basado en funciones de memoria para evitar la repetición de cálculos, como en los esquemas de programación dinámica. Con el fin de reducir costes de procesamiento, se calcula la integral como una suma cuantificada en el espacio de distancias.

El coste de procesamiento se reduce aún más al calcular solo las distancias en parejas que están próximas, de acuerdo a las consultas por series, a los árboles $k$-d instanciados, basadas en los valores de características para las áreas de entrenamiento facilitadas por el usuario.

# Capítulo 5

# Implementación

Procesar volúmenes de datos típicos en teledetección a través de plataformas web requiere integrar estas capacidades en sistemas escalables que permitan obtener resultados en unos tiempos aceptables por parte del usuario (en muchos casos cercanos al tiempo real).

Mientras el capítulo anterior describe la algoritmia propuesta para modelar la supervisión, basado en métodos probabilísticos para el modelado de la incertidumbre, en éste se detalla la metodología de su implementación y las arquitecturas desarrolladas que, como resultado, conforman un servicio web.

En primer lugar, en la sección 5.1 se presenta la implementación del prototipo realizada a nivel local. A continuación, en la sección 5.2 se presenta la gestión de grandes volúmenes de datos de teledetección con metodologías propias del paradigma *Big Data*. La sección 5.3 traslada el desarrollo algorítmico presentado en el capítulo anterior a la computación distribuida en la nube.

## 5.1. Implementación del prototipo desarrollado

La implementación del prototipo pre-operacional para el mapeo temático sobre imágenes de teledetección de muy alta resolución requiere modelar el entrenamiento para cada caso de uso particular. Esto supone la utilización de un modelo algorítmico de carácter adaptativo que no requiera un tiempo de entrenamiento y test excesivo, teniendo en cuenta que el tiempo máximo de espera para la descarga de una página

web es de 40 segundos [8], el tiempo total de procesamiento debe acercarse a esos valores.

El acceso a la implementación a través de un entorno web implica consideraciones sobre usabilidad, escalabilidad y efectividad; así como sobre el tiempo de respuesta, que son habituales en servicios web, que deberán mantenerse para proporcionar una calidad de servicio.

A continuación, se describe la arquitectura empleada en la implementación y el flujo de proceso de trabajo diseñado para un caso de uso concreto.

### 5.1.1. Aprendizaje automático en servidores web

La arquitectura propuesta responde a la clásica arquitectura cliente-servidor, siendo este último la parte donde se ha concentrado la mayor parte del trabajo realizado, dejando para el cliente la interfaz de usuario con la que interactuar con el sistema.

El servidor se puede dividir en tres módulos principales que describiremos a continuación: módulo servidor de mapas, módulo servidor para procesamiento y un módulo servidor web tradicional.

El módulo servidor de mapas basado en *TileStache* [1] gestiona las imágenes, en este caso cuadrículas de imágenes de teledetección, que utiliza el sistema tanto para la visualización como para procesarlas según el entrenamiento proporcionado, o para gestionar los mapas temáticos creados por el sistema. El sistema genera un mosaico temático a diferentes niveles de resolución, basado en las entradas proporcionadas por el usuario. El sistema de clasificación esta acoplado al sistema generador del mosaico para optimizar el tiempo de respuesta.

El módulo de procesamiento es el encargado de procesar las imágenes y la clasificación. Procesa los árboles $k$-d necesarios y ejecuta la clasificación basándose en los polígonos seleccionados por el usuario. El módulo de procesamiento recibe las peticiones de procesamiento desde el módulo del cliente, las procesa y provee las cuadrículas al módulo de mapas. Al mismo tiempo, provee un identificador correspondiente al proceso realizado por el cliente que permite solicitar las cuadrículas que componen el nuevo mapa temático.

El lado cliente se compone de una interfaz gráfica basada en estándares web. Esta

[1] http://tilestache.org/

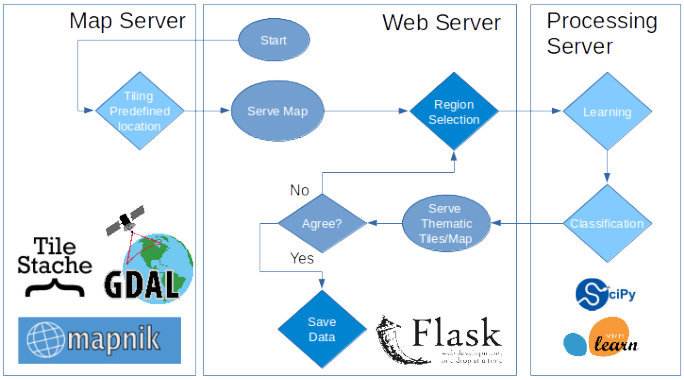



Figura 5.1: Arquitectura implementada en el lado del servidor

interfaz, como se puede ver en la Figura 5.2, está construida alrededor de una vista con un mapa interactivo que soporta el entrenamiento supervisado de acuerdo con la semántica de la clase temática de interés. El panel de configuración presenta una descripción del entrenamiento proporcionado y permite al usuario ajustar algunos de sus parámetros del modelo de aprendizaje de manera interactiva. La interacción está gestionada mediante eventos manejados por librerías jQuery.

### 5.1.2. Flujo de proceso y optimización mediante árboles $k$-d

En un entorno web, la optimización del rendimiento de las tareas relacionadas con la comunicación de datos y el consumo de memoria en el cliente es de vital importancia. En el caso en que los datos tengan un volumen que permita almacenarlos en la memoria de un solo servidor, [105], [106], se pueden calcular árboles $k$-d estáticos. En el caso de que el volumen de datos de la capa obstaculice una gestión ágil, se necesita una estrategia dinámica.

La solución desarrollada trata de ser simple y efectiva, creando sólamente los árboles $k$-d necesarios. La capa creada está limitada al área disponible alrededor del área visible en el navegador. Esta estrategia requiere una mayor comunicación entre el cliente y el servidor para que este último genere y procese los árboles $k$-d necesarios. A medida que el usuario navega por el mapa, el cliente envía la información relacionada con el área de visualización para poder actualizar la visualización de acuerdo con el procesamiento realizado en base al entrenamiento proporcionado.

Así, el servidor es capaz de crear los árboles $k$-d relacionados con la navegación

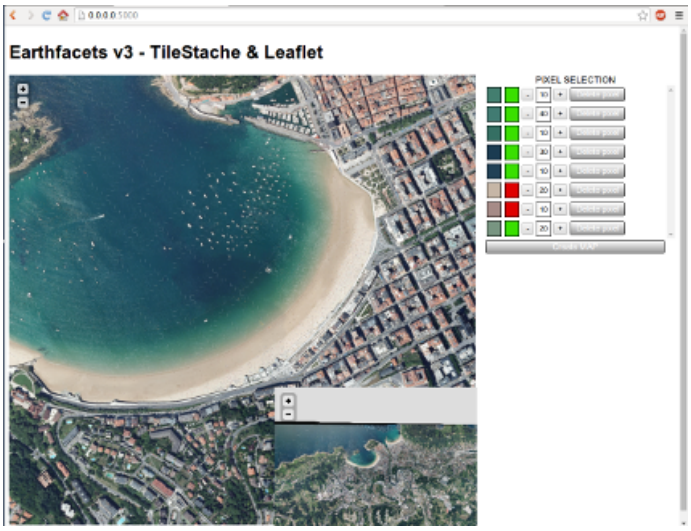



Figura 5.2: Interfaz de usuario. El visor de mapas proporciona herramientas para generar el entrenamiento supervisado y la representación de los resultados de salida. El panel de configuración de la derecha permite al usuario manipular de forma interactiva los parámetros para del modelo de clasificación supervisada en base al entrenamiento proporcionado.

y procesarlos con la información recibida . Eventos que impulsan una extensión o recálculo del área activa o analizada se gestionan generando nuevas peticiones al servidor. La configuración del sistema tiene como objetivo reducir estas peticiones al mínimo, evitando al mismo tiempo una carga excesiva en la memoria del cliente.

El usuario supervisor es libre de definir una clase semántica basada en una composición probabilística de componentes simples, cada uno representado mediante una instancia de árbol $k$-d diferente.

Un ejemplo del funcionamiento del sistema para una petición del cliente para la creación de un mapa temático es la secuencia representada en el Figura 5.3. La descripción de los pasos es la siguiente:

1. El proceso comienza renderizando un mapa de un área predefinida por parte del servidor de mapas.

2. A continuación, el mapa se divide mediante una cuadrícula, como si fuera un mosaico de donde se calculan los árboles $k$-d para cada parte de la cuadrícula del mapa para procesar el entrenamiento.

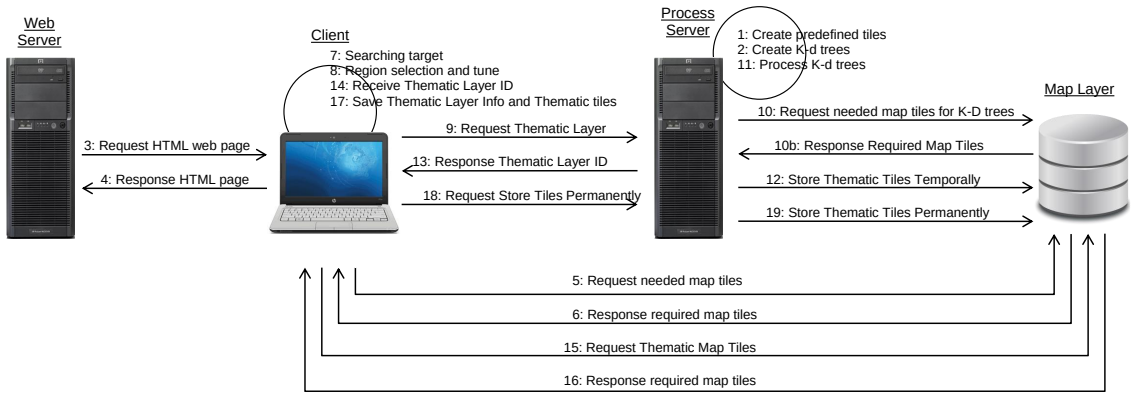



Figura 5.3: Diagrama cliente/servidor de secuencia de formación de mapas temáticos.

3. El cliente solicita una página web al servidor web.

4. El servidor web recibe la solicitud y responde con una página web HTML con la información necesaria para crear el mapa.

5. El cliente comienza a navegar por el mapa, solicitando las piezas de la cuadrícula necesarias para visualizar el mapa al servidor de mapas.

6. El servidor de mapas genera las piezas solicitadas y el cliente las carga. En este punto, el cliente está preparado.

7. A continuación, el usuario busca en el mapa las instancias que caractericen la clase objetivo.

8. El usuario selecciona regiones de acuerdo a la semántica de su búsqueda, pudiendo ajustar los parámetros de configuración del modelo.

9. Una vez el usuario ha finalizado la fase de selección o caracterización de su búsqueda, se solicita una nueva capa del área activa o visualizada. Las regiones seleccionadas se convierten en el entrenamiento y, junto con su configuración, son enviados vía peticiones asíncronas AJAX, evitando la espera de la respuesta.

10. Cuando el servidor de procesamiento recibe los datos, comprueba si los árboles $k$-d necesarios han sido creados o no, para solicitar las piezas necesarias del

mapa al servidor de mapas.

11. Con los árboles $k$-d creados, se procesan los datos de entrenamiento. Este proceso crea una nueva cuadrícula para cada árbol compuesta por los píxeles vecinos más cercanos a la clase de entrenamiento correspondiente.

12. Las nuevas cuadrículas son almacenadas en el servidor de mapas.

13. Una vez todas las nuevas cuadrículas han sido creadas, se devuelve un identificador desde el servidor de procesamiento al cliente.

14. El cliente recibe el identificador, pudiendo activar o desactivar el nuevo mapa en forma de capa.

15. El cliente solicita la nueva capa al servidor de mapas

16. El servidor de mapas devuelve las cuadrículas necesarias para crear la nueva capa. El cliente visualiza la nueva capa sobre el mapa original.

17. El cliente puede guardar la capa temática creada, guardando la información de entrenamiento y configuración de forma local.

18. El cliente puede lanzar una petición de guardado.

19. El servidor de procesamiento recibe la petición y lo solicita al servidor de mapas.

## 5.2. Gestión de grandes volúmenes de datos de teledetección. Adaptación a entornos *Big Data*

En la Figura 5.4 se presenta un esquema funcional de la idea propuesta. El sistema implementado puede verse como un catálogo de cuadrículas de una cobertura particionada. Cada cuadrícula se describe como una entidad completamente independiente en términos de metadatos, así como por la imagen raster correspondiente a la cobertura original sobre la que se ejecutará el proceso de análisis.

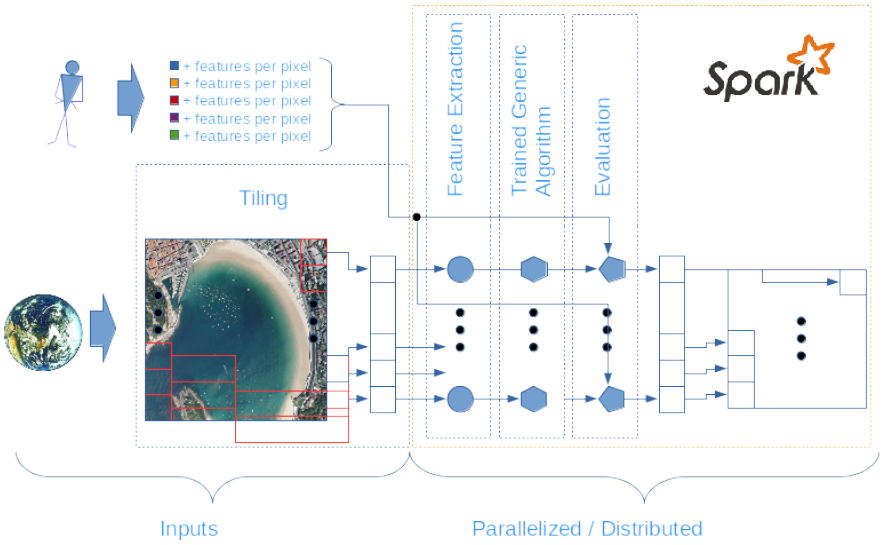



Figura 5.4: Diagrama de arquitectura de alto nivel del flujo de procesamiento implementado. La generación de la cuadrícula para coberturas *raster* a gran escala permite la distribución independiente de los datos sobre los nodos de un cluster de computación en memoria para *Big Data* del tipo de Apache Spark. En cada nodo se realiza la extracción de características y su clasificación supervisada basada en el modelo distribuido y previamente entrenado con un limitado conjunto de datos.

Seleccionada el área geográfica que hay que analizar y las fuentes de datos disponibles, el área geográfica se divide en una cuadrícula con un tamaño predefinido de arista para poder distribuir fácilmente el análisis sobre el entorno de procesamiento completo. En este punto, las imágenes se convierten a un formato *RDD* [78].

Los datos de entrenamiento proporcionados por un supervisor humano modelan el aprendizaje automático mediante un algoritmo clasificador que se distribuye entre los nodos del cluster.

Una vez el algoritmo se ha distribuido, se evalúan los datos generando como resultado nuevas cuadrículas basadas en los resultados de la clasificación.

Finalmente, todas las cuadrículas vuelven a fusionarse en una única capa a modo de resultado final para poder contrastar visualmente el resultado obtenido con el original. En caso de existir algún tipo de información de validación, ésta se podría confrontar con los resultados obtenidos para obtener medidas estadísticas del rendimiento de la clasificación.

El prototipo implementado es instanciado y ejecutado en la infraestructura como servicio *IaaS*, de sus siglas en inglés *Infrastructure as a Service*, de *Amazon Elastic Computing Cluster*, con el fin evaluar adecuadamente las opciones de escalabilidad (horizontal y vertical del sistema). Esta disponibilidad también permite evaluar experimentalmente los parámetros de particionamiento de la cuadrículas para el análisis de una configuración de una específica.

La modularidad ofrecida por Apache Spark permite aplicar fácilmente diferentes algoritmos de clasificación y agrupamiento mediante una interfaz estandarizada.

## 5.3. Distribución de *T-P-kNN* optimizado en entornos *Big Data*

En la sección 3.4 se introducen los diferentes mecanismos existentes para procesar eficientemente grandes volúmenes de datos en el ámbito de la teledetección. En esta sección se describe el enfoque de procesamiento paralelo para el análisis de imágenes de teledetección, basado en el paradigma *MapReduce* e implementado en el entorno Apache Spark.

El algoritmo presentado en la sección anterior, Sección 4.1, está basado en una implementación en serie de un algoritmo de clasificación supervisada. El algoritmo *per se*, como es habitual en el aprendizaje automático, es inherentemente paralelizable, aunque necesita ser revisado para poder gestionar grandes volúmenes de datos de manera eficiente.

Debido a que el algoritmo está implementado mediante un lenguaje interpretado o de *scripting*, el tiempo de procesamiento necesario para realizar una clasificación supervisada sobre una imagen de 25 Megapixeles se sitúa en torno al minuto con un equipo provisto de un procesador Intel® Core™ i7-4750HQ (2.0GHZ), 8 GB de memoria RAM y disco duro SSD. Si extrapolamos estos valores a extensiones regionales, por ejemplo al País Vasco con un mapa sobre los 150 Mpíxeles, el resultado en tiempo de procesamiento se eleva hasta las 20 horas de procesamiento.

La solución propuesta pasa por explotar las capacidades de procesamiento distribuido sobre datos cartográficos divididos en cuadrículas, a partir de un modelo de aprendizaje automático basado en la clasificación supervisada bajo el paradigma

*MapReduce*.

La extracción de características completamente independiente en cada una de las cuadrículas puede expresarse mediante funciones tipo *Map* ejecutadas en cada nodo del entorno de procesamiento. La clasificación supervisada puede hacerse efectiva en un subconjunto pequeño de datos de entrenamiento, sobre el 1 %, de los datos a gran escala. El modelo clasificador resultante puede ser distribuido a cada uno de los nodos y evaluado sobre las características extraídas en una operación perfectamente paralela.

La función *Reduce*, no siempre necesaria, puede describirse como un fase de fusión, implementada mediante la recolección de los datos distribuidos.

Como se describe en la sección 3.4, la implementación dominante de *MapReduce* hasta este momento es el entorno *open source* Hadoop de la Apache Foundation, caracterizado en gran medida por el uso del disco duro de las maquinas como unidad de almacenamiento durante los procesos de cálculo. En este caso, se ha optado por implementar el prototipo mediante Apache Spark, una implementación alternativa optimizada para trabajar en memoria RAM. El prototipo se ha implementado como servicio de Amazon Elastic Computing Cluster con la intención de evaluar las diferentes opciones de escalabilidad, tanto vertical como horizontal, de las que dispone el sistema. La escalabilidad vertical se resuelve ampliando las capacidades hardware de los sistemas, mientras que la horizontal se resuelve añadiendo más nodos de procesamiento al sistema. Esta particularidad también nos permite evaluar experimentalmente la optimización de la cuadrícula de división de la superficie que vamos a analizar para una configuración específica.

La modularidad proporcionada por Apache Spark permite evaluar diferentes algoritmos de clustering y clasificación mediante una interfaz estandarizada.

# Capítulo 6

# Validación experimental

Este capítulo describe las metodologías de evaluación y los resultados de su aplicación sobre las diferentes implementaciones realizadas. El capítulo se divide en dos partes: la metodología seguida para la validación y la presentación de los resultados obtenidos. La metodología está compuesta por el desarrollo de mapas de validación para la validación de los resultados -Sección 6.1-, y la metodología utilizada para la evaluación -Sección 6.2-. La presentación de los resultados está compuesta por los resultados sobre el rendimiento de la clasificación -Sección 6.3-, y los resultados referentes al tiempo de procesamiento -Sección 6.4-.

## 6.1. Desarrollo del entorno de validación mediante imágenes de teledetección

La validación de los resultados de la clasificación es una tarea imprescindible a la hora de evaluar el funcionamiento del sistema, tal y como se comentó en la sección 2.1. El caso de la clasificación sobre imágenes de teledetección no es una excepción y se requieren sistemas y métodos con los que poder evaluar los resultados obtenidos.

En nuestro caso, la evaluación del sistema pasa por contrastar los resultados obtenidos con mapas de evaluación que cubran la superficie procesada. La comparación de los resultados permite completar la matriz de confusión, de la que se pueden obtener medidas estadísticas con las que medir el rendimiento del sistema, en base a funciones relacionadas con la recuperación de la información como las descritas en

la Sección 2.5.

Para ello, se ha tenido que desarrollar un sistema para la creación de mapas de validación a partir de datos públicos compuestos de mapas vectoriales e imágenes aéreas que nos permita evaluar la clasificación en términos estadísticos.

En este caso, se ha recurrido a los datos disponibles en el portal de acceso a datos públicos del Gobierno Vasco, *Open Data Euskadi* [1] . El objetivo principal de esta iniciativa es generar valor y riqueza a partir de los datos, mediante el desarrollo de productos derivados de ellos. Este portal dispone de un apartado de datos geográficos denominado *geoEuskadi* [2], además de información relativa a la calidad del agua y del aire, el patrimonio y recursos culturales, normas y leyes, contrataciones, oferta pública de empleo y ayudas y subvenciones.

Este apartado dispone de su propio portal y, en él, se pueden encontrar diferentes catálogos de datos y servicios distribuidos en categorías como son la agricultura, la cartografía, la biología y el medio ambiente, por citar algunas de ellas. Todas las categorías disponen de su correspondiente representación en forma de mapa, compuesta por varias subcategorías y diferentes niveles. Estas categorías, a su vez, están definidas por diferentes atributos en forma de metadatos. Todas estas categorías están disponibles en el visor web integrado del que dispone el portal.

De los datos disponibles en el repositorio de Open Data Euskadi, en este trabajo de se han utilizado las ortofotos y mapas vectoriales disponibles. Mientras que las ortofotos se han utilizado como base para la modelización de los algoritmos implementados durante el proceso y su posterior evaluación, los mapas vectoriales se han utilizado para la generación de mapas de validación, para poder evaluar el rendimiento de los algoritmos mediante medidas estadísticas.

### 6.1.1. Generación de mapas de validación mediante bases de datos de ortofotografías de gran extensión

Tal y como define el Institut Cartogràfic i Geològic de Catalunya [3] , una ortofoto es un composición de *"fotografías aéreas que han sido rectificadas para adaptarse a*

[1] http://opendata.euskadi.eus/

[2] http://www.geo.euskadi.eus/

[3] http://www.icc.cat/esl/Home-ICC/Mapas-escolares-y-divulgacion/Preguntas-y-respuestas/Diferencias-entre-fotografia-aerea-y-ortofoto

*la forma del terreno, de tal forma que el punto de vista de la cámara no afecte a la posición real de los objetos".*

A diferencia de una imagen aérea, en una ortofoto pueden realizarse mediciones reales, ya las correcciones de las distorsiones inherentes a las imágenes aéreas la convierten en una representación precisa de la superficie terrestre. Las ortofotos disponen de la precisión geométrica de los mapas a escala uniforme y características de detalle y cobertura temporal de las fotografías aéreas. Gracias a esto, se pueden superponer diferentes elementos de los mapas sobre ellas.

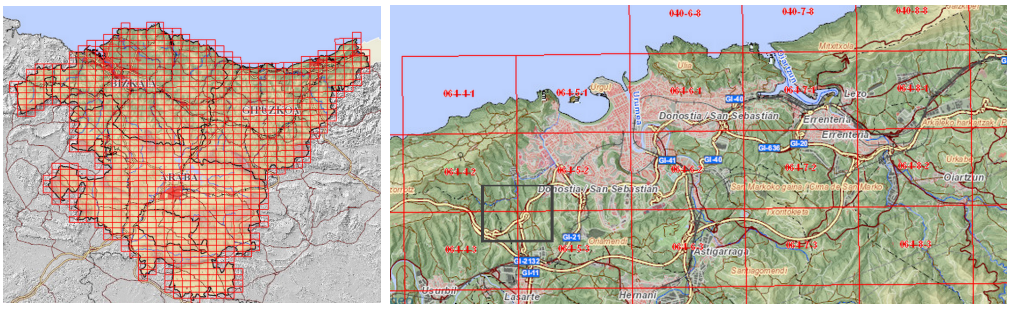
Según la indicaciones de la web geoEuskadi, las ortofotografías disponibles cuentan con una resolución de 25cm por píxel con radiometría RGB. Se han publicado en el sistema geodésico de referencia ETRS89 y coordenadas UTM, en formato ECW para toda la CAPV y por municipios; y en JPEG por hojas, en las escalas 1:5000, 1:10000, 1:25000 y 1:50000. Por otro lado, en el Servicio de Información Territorial de la Dirección de Planificación Territorial y Urbanismo del Departamento de Medio Ambiente y Política Territorial del Gobierno Vasco, se pueden encontrar la serie de hojas 1:5000 con las bandas RGBIr en formato TIFF y remuestreos de la ortofotografía en RGB.

La selección de las localizaciones para la evaluación está orientada a través de las categorías seleccionadas en los mapas de evaluación. Por otro lado, el procesamiento ideado posibilita dos modelos de selección de conjuntos de validación: la selección de múltiples regiones con una sola clase o la selección de amplias regiones con múltiples clases. Aunque de mayor complejidad, en este caso se considera la segunda, excluyendo la primera por considerarla poco realista y por la dificultad que conlleva seleccionar múltiples regiones que conformen un clase. Un ejemplo de esta dificultad sería seleccionar diferentes tipos de bosques de diferentes especies.

En primer lugar, se han de localizar las hojas cartográficas en las que estén representadas las clases seleccionadas, entre las hojas que representan el territorio que se va a analizar, tal y como aparecen en la Figura 6.1 a). Una vez localizadas, generan las imágenes correspondientes, atendiendo a las limitaciones sobre la extensión de las imágenes resultantes, a partir de dichas hojas.

Mediante librerías de procesamiento geoespacial, como *GDAL* [4] , se generan las

[4] http://www.gdal.org



(a) (b)

Figura 6.1: Malla cartográfica utilizada para la distribución de las imágenes aéreas de Euskadi (a). Detalle de la malla cartográfica sobre la ciudad de Donostia-San Sebastián en la que se aprecia el esquema de distribución utilizado.

imágenes que forman el conjunto de evaluación, tras la conversión de coordenadas geoespaciales a píxeles, que delimitan el área seleccionada.

El conjunto de evaluación considerado está compuesto por 5 localizaciones diferentes del País Vasco, véase la Figura 6.2, cada una con una extensión de 1,2 km por 1,2 km, que se traduce en unos 4.864 por 4.864 píxeles. La composición de las localizaciones se puede apreciar en la Figura 6.3.

El análisis se lleva a cabo sobre 25000 × 5000 píxeles, con una resolución correspondiente a 25 cm por píxel. Como es típico de los sistemas de adquisición de imágenes con muy alta resolución geométrica, la resolución radiométrica de los datos adquiridos está limitada respecto al número de canales disponibles y a la resolución geométrica.

Las ubicaciones incluyen los 12 tipos diferentes de coberturas identificadas, correspondientes a atributos de mapas geográficos de referencia extraídos del repositorio de Open Data Euskadi. Las coberturas identificadas son: *Arbustos*, *Área industrial*, *Área urbana*, *Bosques*, *Campos*, *Carreteras*, *Edificios*, *Mar*, *Masas de agua*, *Pastos*, *Playas* y *Suelo desnudo*.

### 6.1.2. Validación frente a mapas vectoriales

En la Figura 6.4 se presenta un diagrama que muestra en el proceso de transformación realizado a los *shapefiles* para la generación del mapa de validación vectorial.

Una vez analizadas las diferentes capas semánticas que pueden ser seleccionadas

| Id | Nombre localización | Lat/Lon | Descripción localización | Coberturas clases |
|---|---|---|---|---|
| 1 | La Concha | 43.3190, -1.9923 | Bahía | Suelo desnudo, playa, campos, carreteras, mar, edificios, bosques y zona urbana. |
| 2 | Goroeta | 43.0056, -2.4737 | Montaña | Suelo desnudo, playa, campos, pastos, carreteras, edificios, arbustos y bosques. |
| 3 | Barakaldo | 43.2986, -3.0004 | Área industrial | Edificios, zona industrial, zona urbana, carreteras y masas de agua. |
| 4 | Vitoria Gasteiz | 42.8505, -2.6690 | Zona urbana diversa | Edificios, carretera y zona urbana. |
| 5 | Urdaibai | 43.3837, -2.6905 | Estuario/ reserva natural | Edificios, campos, pastos, zona urbana, masas de agua, carreteras y bosque. |

Tabla 6.1: Localizaciones de las áreas de test utilizadas en la evaluación. Las cinco localizaciones representan un significativo grado de diversidad contextual así como un significante número de clases de cobertura específicas. El área de test está delimitada por la latitud y longitud de las coordenadas del centro de las imágenes dadas y el tamaño de la imagen resultante limitada a 5000 x 5000 píxeles. Cada una de las cinco localizaciones tiene una extensión aproximada de 1,2 $\times$ 1,2 $km^2$, que corresponde a unos 23,6 Mpíxeles, haciendo un total de unos 150 Mpíxeles.

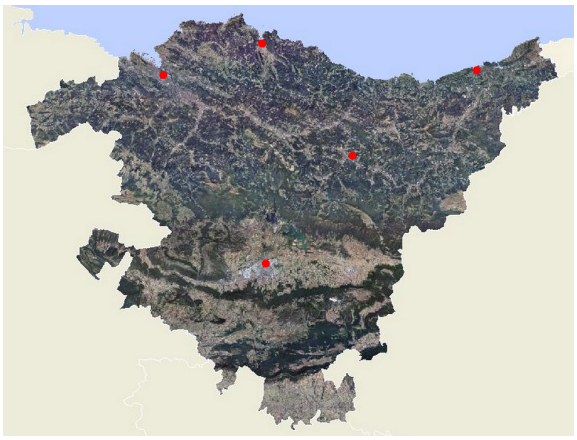


Figura 6.2: Localización geográfica de los emplazamientos para la evaluación. De izquierda a derecha y de arriba abajo, zona industrial a las afueras de Bilbao, el área protegida del estuario de Urdaibai, la bahía de La Concha, zona montañosa en Goroeta y entorno urbano diverso en Vitoria-Gazteiz. Estas áreas incluyen las 12 clases de cobertura consideradas: playa, edificios, campos, zona industrial, suelo desnudo, pastos, arbustos, mar, zona urbana, carreteras y masas de agua. En la Tabla 6.1, se presentan las características de las localizaciones seleccionadas.

como posibles categorías, se obtienen los archivos *.shp* necesarios, junto con sus metadatos. Los metadatos que cargamos junto a las capas son de gran ayuda, ya que podemos filtrarlos, agruparlos, crear nuevos, etc., y se convertirán en elementos esenciales a lo largo del proceso.

Las capas y los metadatos se cargan en la aplicación y comienza la caracterización del mapa de validación. Una vez seleccionadas las categorías que cubren las necesidades de la implementación, hay que asegurarse de que las capas seleccionadas cubran toda la extensión que supone el mapa de validación para que no queden píxeles sin definir.

A partir de este punto, el proceso y las operaciones que hay que realizar en cada capa pueden variar, dependiendo de los procesos necesarios para lograr el mapa de validación requerido en cada caso.

Es por ello que únicamente se comentarán las operaciones realizadas en nuestro caso, amén de reflejar el proceso llevado a cabo para obtener el mapa de validación.

Los metadatos también han servido para seleccionar el nivel de categoría requerido en cada caso. Por ejemplo, en la capa *Cartografía Básica*, los edificios se han

(a) (b)

(c) (d)

(e) (f)

Figura 6.3: Composición de las 5 imágenes originales de evaluación. De izquierda a derecha, la bahía de La Concha, zona montañosa en Goroeta, zona industrial a las afueras de Bilbao, entorno urbano diverso en Vitoria-Gazteiz y el área protegida del estuario de Urdaibai. En 6.3f detalle 1:1 de la imagen 6.3d original que permite apreciar la resolución sub-métrica de los datos considerados en el análisis.

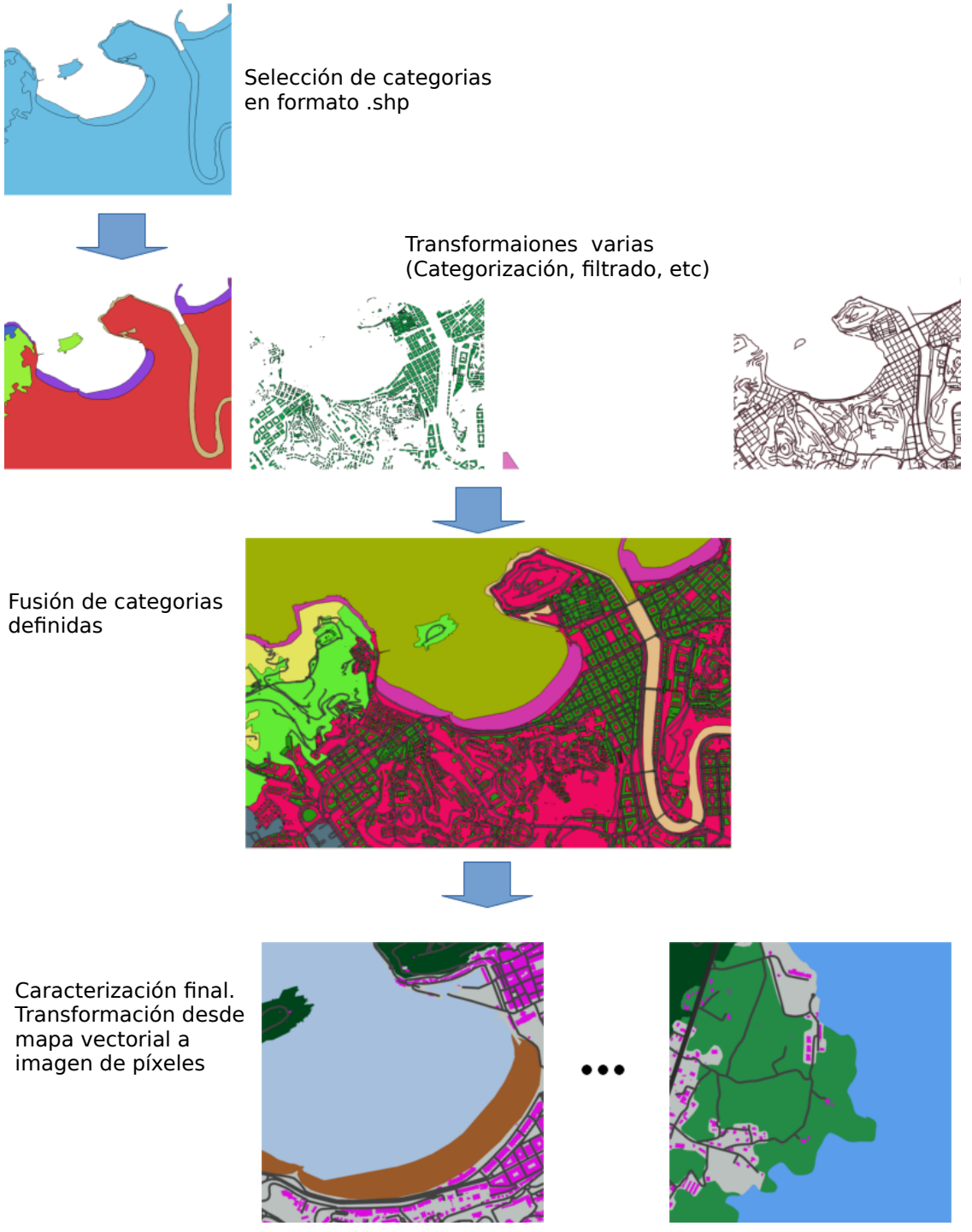



Figura 6.4: Diagrama de proceso de generación de un mapa vectorial. Después de la selección de las categorías, es necesario un proceso que modifique las diferentes capas, para adecuarlas a las necesidades de cada caso. A continuación, se rasterizan las categorías, para finalmente fusionar las imágenes resultantes en un solo mapa.

agrupado bajo la etiqueta *Edificaciones* sin importar si son naves industriales o iglesias. También han servido para agrupar los objetos bajo etiquetas específicas, más acordes con la descripción necesaria para nuestra implementación que la original.

Una vez seleccionadas las capas, los niveles de cada una de ellas y los pertinentes ajustes de los parámetros; la capas se deben ordenar de forma que los diferentes objetos no se solapen o queden completamente cubiertos por otras capas.

Ordenadas las capas, éstas se fusionan en una sola. Para ello, en primer lugar se deben rasterizar a modo imagen. Esta operación se realiza mediante la librería *GDAL*, librería que sirve de puente entre datos vectoriales, las capas de las diferentes categorías y las imágenes *raster*. Si no se indica lo contrario, el proceso exportará toda la capa a modo imagen, lo que en nuestro caso sería toda la extensión del País Vasco. En este caso particular, se definen los centros de las localizaciones, de donde queremos generar los mapas de validación mediante los parámetros de latitud y longitud y las dimensiones de la imagen resultante en píxeles.

En la exportación a modo imagen, las informaciones se fusionan en el orden correspondiente para, finalmente, obtener una imagen del mapa de validación vectorial.

### 6.1.3. Problemática sobre mapas vectoriales

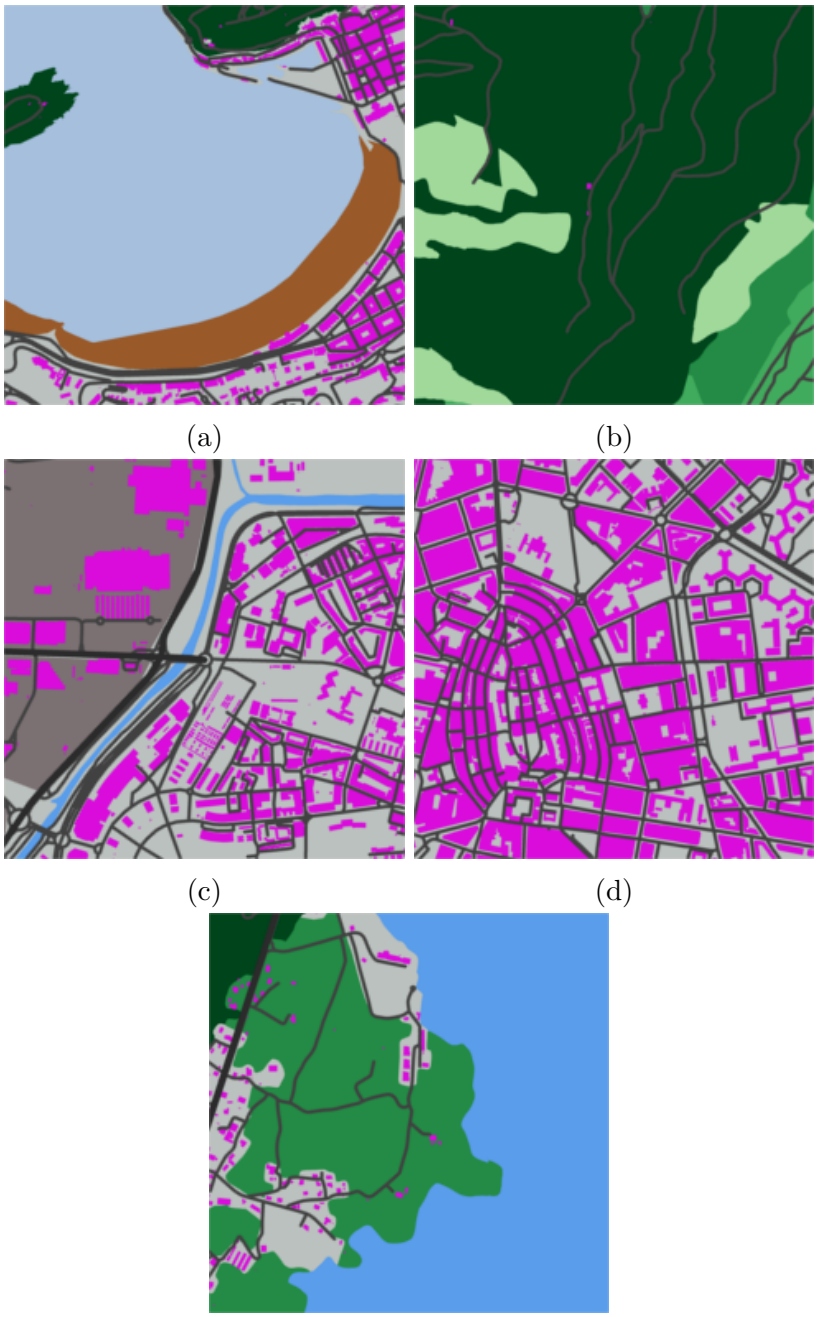
Como se ha visto, generar un mapa de validación que cubra el área que se va a analizar es un proceso complejo y extenso en el tiempo. Obtener un mapa de validación lo más descriptivo posible requiere fusionar las diferentes categorías de capas existentes, cubriendo todas y cada una de necesidades de cada implementación.

El proceso se debe realizar manualmente para una correcta gestión de las diferentes categorías semánticas dado que, entre otras cosas, algunas categorías se superponen entre sí y otras no cubren adecuadamente las áreas que se van a analizar.

Los mapas de validación obtenidos para las áreas de análisis, presentados en la Figura 6.3, se presentan en la Figura 6.5.

Solucionado el problema de la diferencia semántica entre mapas y categorías, surgen dos tipos de desalineaciones entre los mapas vectoriales disponibles y las correspondientes imágenes, una temporal y otra espacial.

El primer problema es el instante de referencia de los mapas respecto a la adquisición de las imágenes. Un claro ejemplo de este problema puede observarse en



(a) (b) (c) (d) (e)

Figura 6.5: Composición de los mapas de validación correspondientes a las áreas de evaluación presentadas en 6.3, de izquierda a derecha y de arriba hacia abajo, la bahía de La Concha, zona montañosa de Goroeta, zona industrial a las afueras de Bilbao, zona urbana de Vitoria-Gazteiz y el área protegida de Urdaibai. Mapas de validación basados en *shapefiles*, obtenidos en el portal WMS de Open Data Euskadi.

el efecto producido por la marea baja en la desembocadura del río en el estuario de Urdaibai, donde los mapas vectoriales se representan considerando la subida de la marea.

El segundo problema se refiere al nivel de detalle de los mapas vectoriales donde, típicamente, no coinciden a nivel de píxel con las imágenes aéreas debido a la resolución de 25 cm por píxel de estos últimos. Ambas desalineaciones tendrán un claro efecto negativos a la hora de medir el rendimiento del sistema. Si se comparan las Figuras 6.3 y 6.5, es sencillo detectar algunas diferencias: la mayoría de las zonas verdes en las zonas urbanas no están representadas y diferentes tipos de vegetación en la zona de Goroeta están catalogadas bajo la misma etiqueta.

Para hacer frente a estas limitaciones, además del mapa de validación basado en mapas vectoriales, se ha considerado la creación de un mapa de validación a nivel de píxel, para poder comparar con mayor detalle los resultados obtenidos.

### 6.1.4. Validación frente a mapas detallados

La imposibilidad de dedicar los recursos necesarios para desarrollar manualmente los diferentes mapas de validación, nos ha obligado a considerar una única localización entre las 5 localizaciones presentadas anteriormente. La localización seleccionada ha sido la bahía de La Concha en la ciudad de Donostia–San Sebastián. La bahía reúne el mayor número de coberturas diferentes en una limitada extensión, lo que la hace particularmente interesante.

El primer paso es la selección del conjunto de clases semánticas con un claro significado dentro de la imagen. En este caso, se han seleccionado 8 clases, *playa, edificios, suelo desnudo, jardines, carreteras, mar, bosques y zona urbana*. Aunque estas clases solo representan una aproximación de las 12 clases consideradas en el caso del mapa de validación basado en mapas vectoriales, consideramos que el conjunto es lo suficientemente significativo, ya que las clases representan adecuadamente los contenidos visibles esenciales y la separación semántica es *a priori* suficientemente.

Dentro de este estudio, se requiere un repaso previo del mapa, con el que unificar criterios, frente a situaciones que no se hayan contemplado en un principio. Sirva por ejemplo la Figura 6.3f donde, debido a la hora en que se capturó la imagen, los edificios proyectan una sombra sobre los diferentes elementos aledaños, abarcando

una superficie considerable.

Si para el desarrollo de mapas de validación vectoriales el uso de entornos *GIS* era lo más indicado, en el caso de los mapas de validación detallados a nivel de píxel, los entornos de edición gráficos *rasterizados* son de gran utilidad.

Una vez se han definido múltiples mapas temáticos uniclase, se requiere un proceso para fusionarlos en un mapa multiclase. Mediante un proceso semiautomático, se destacan las áreas asignadas a múltiples clases, así como las no asignadas. Los píxeles de estas áreas están sujetos a un procedimiento de arbitraje automático, con el fin de asignarlos inequívocamente, a una única clase temática dentro del mapa de validación.

El mapa de validación generado, presentado en la parte derecha de la Figura 6.6, se publica actualmente como open data. [5].

### 6.1.5. Caracterización de los datos

Según lo indicado en el estado del arte sobre la caracterización de imágenes para aplicaciones de teledetección [80], excepto los descriptores de color RGB y HSV, las características primitivas consideradas están basadas en regiones: histogramas de Gradientes Orientados (HOG), patrones binarios locales (LBP), detector de ángulo recto / detector de segmento de línea (LSD), la densidad de bordes y SIFT.

En este trabajo, se ha establecido un proceso para la extracción de los descriptores que define una rejilla común entre los descriptores extraídos, a fin de facilitar un procedimiento de fusión de datos. Esto requiere un reescalado a resolución espacial común como una interpolación del vecino más cercano para los descriptores de menor resolución espacial. Las descripciones de las características primitivas con los parámetros de extracción correspondientes, incluyendo el tamaño de la región, se presentan en la Tabla 6.2

Como en [80], el entrenamiento supervisado es proporcionado al sistema en forma de regiones poligonales delimitadas manualmente y definidas sobre superficies con una clase específica en la imagen de entrada (ver Tabla 6.3). Se realiza un muestreo sin reemplazo para extraer un número igual de muestras, por lo general en el orden de decenas de miles para todas las clases de entrenamiento. Los conjuntos de muestras

[5] http://150.241.250.4:5000/earthfacets/groundtruthmap.png

(a)

(b) (c)

Figura 6.6: De izquierda a derecha: imagen original (a), mapa de evaluación basado en mapas vectoriales (b), mapa de evaluación a nivel de pixel realizado a mano basado en la interpretación de la imagen de la localización de la bahía de La Concha (c). El subconjunto de clases de cobertura definido incluye 8 clases de las 12 originales: edificios (morado), mar(azul), suelo desnudo (verde claro), pastos (verde), bosques(verde oscuro), zona urbana(gris), carreteras(gris oscuro) y playa (marrón). El incremento del nivel de detalle es evidente, especialmente en áreas con vegetación entre edificios y áreas con suelo desnudo.

| Nombre descriptor | Tamaño región | Cuantificación angular | Referencia |
|---|---|---|---|
| Edge Density | $12 \times 12$ | Ninguna | [99] |
| HOG | $12 \times 12$ | 8 | [107] |
| LBP | $24 \times 24$ | 8 | [107] |
| LSD | $12 \times 12$ | 4 | [100] |
| SIFT Density | $24 \times 24$ | Ninguna | [100] |

Tabla 6.2: Descriptores geométricos de contenidos de imagen con parámetros de extracción utilizados en el presente trabajo.

extraídos se utilizan para estimar las *PDF* para distribuciones específicas de las clases.

## 6.2. Metodologías de evaluación

En esta sección se describen las metodologías utilizadas para evaluar los prototipos implementados. Esta sección está dividida en dos partes. En primer lugar, se presenta una metodología de evaluación para la medición del rendimiento del aprendizaje automático mediante la clasificación supervisada, tanto en local como en el caso de la implementación en la nube. En segundo lugar, se presenta la metodología llevada a cabo para la medición del rendimiento del tiempo de procesamiento, con dos enfoques diferentes. Mientras que, en un escenario local, se estudia el rendimiento ofrecido por diferentes motores de indexación de datos; en un entorno preparado para la nube, se estudia cómo afecta al rendimiento el tamaño de los bloques de datos que hay que distribuir.

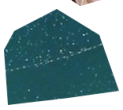
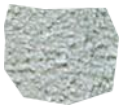
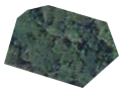
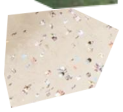
| Clase | Región de entrenamiento | Clase | Región de entrenamiento |
|---|---|---|---|
|  | Arbustos |  | Edificios |
|  | Área industrial |  | Mar |
|  | Área Urbana |  | Agua |
|  | Bosques |  | Pastos |
|  | Campos |  | Playas |
|  | Carreteras |  | Suelo desnudo |

Tabla 6.3: Áreas utilizadas para el entrenamiento. Los píxeles de entrenamiento son muestreados sin reemplazo en un número de 1024 por clase, del global de los píxeles de los polígonos, que corresponden a áreas identificadas en los mapas de validación. De este modo, el conjunto de datos utilizado en el entrenamiento supone $1024 \times 12 = 12288$ pixeles.

### 6.2.1. Metodología de evaluación para la clasificación supervisada

La validación del sistema se centra en la evaluación de principio a fin del mismo, dado que esta operación involucra a todos los subsistemas que componen el prototipo. La evaluación del sistema se lleva a cabo mediante el análisis de la calidad de las imágenes que componen los mapas temáticos basados en una entrada conocida.

La evaluación tiene como base principal la comparación entre dos imágenes, el mapa resultante de la clasificación y un mapa de evaluación.

Ambos mapas de evaluación pueden usarse del mismo modo para evaluar el rendimiento del aprendizaje del prototipo. El proceso empleado para evaluar cuanti-

tativamente los resultados obtenidos no varía en exceso del flujo del procesamiento presentado en la sección 5.1.2. El proceso es el que sigue:

1. Se genera el entrenamiento en forma de polígonos seleccionados por el usuario.
2. Se genera el mapa temático, tal y como se describe en los capítulos 4 y 5
3. Se confronta el mapa obtenido con cualquiera de los mapas de validación para calcular la matriz de confusión.
4. Una vez formada la matriz de confusión, se calculan las estadísticas de rendimiento deseadas.
5. Finalmente, se mide el tiempo necesario requerido por el proceso.

El proceso diseñado permite la evaluación de diferentes modelos de entrenamiento mediante resultados cuantitativos y cualitativos de una manera rápida y sencilla.

### 6.2.2. Medición del rendimiento de la indexación

La eficiencia del aprendizaje respecto al volumen de datos de entrenamiento es de gran importancia en un algoritmo de clasificación supervisada, integrado en un servidor web.

La evaluación de la eficiencia de indexación de los algoritmos se basa en el cálculo de tiempo necesario para evaluar una cantidad de datos limitada con diferentes configuraciones, a fin de optimizar al máximo el rendimiento del enfoque propuesto. La evaluación del rendimiento se realiza tanto en la implementación a nivel local como en la nube.

Aun siendo la localización del procesamiento y sus capacidades distintas en las dos evaluaciones presentadas, se considera oportuno utilizar el mismo conjunto de datos de evaluación para analizar el rendimiento de los algoritmos mediante mediciones estadísticas.

La comparación entre las medidas estadísticas establecidas en la Sección 2.5 da pie a comparar el rendimiento de indexación de los diferentes enfoques.

La validación local se ha centrado en la evaluación de diferentes enfoques de algoritmos de clasificación supervisada basados en vecindad. Se han dispuesto tres casos

con diferentes enfoques sobre el algoritmo de $k$-vecinos más cercanos, sin optimización de indexación, con indexación optimizada mediante árboles $k$-d y con indexación optimizada mediante consultas en bloque sobre árboles $k$-d.

En lo referente a la nube, se ha optado por cuantificar el efecto generado en el procesamiento, debido a la diferente distribución de los datos, sobre el entorno de procesamiento, resultado de la utilización de un tamaño de arista diferente para la creación de la cuadrícula. En este caso, se ha optado por mantener el mismo algoritmo en todas las pruebas realizadas.

## 6.3. Resultados de clasificación

Las características extraídas, a partir de los descriptores presentados en la Sección 3.5 de las imágenes mostradas en la Tabla 6.3, forman el entrenamiento con el que se ha modelado el algoritmo presentado en la Sección 2.4.3 para producir los siguientes resultados.

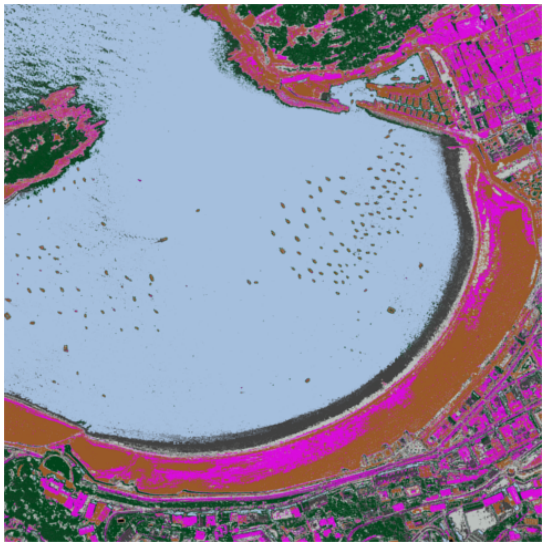
Un ejemplo de resultados es el mapa obtenido mediante un proceso de clasificación con 6 clases basado solamente en descriptores de color presentado en la Figura 6.7b.

Analizando los resultados mostrados en la Tabla 6.4, los valores de exactitud rondan el 85 por ciento en la mayoría de las clases. A diferencia de los edificios del casco histórico, bien clasificados gracias a sus tejados construidos con tejas, las edificaciones más modernas se caracterizan por un menor rendimiento debido a la variedad de patrones de los tejados. La clase *Área urbana* es otro caso en donde la diversidad de patrones dificulta una buena caracterización.

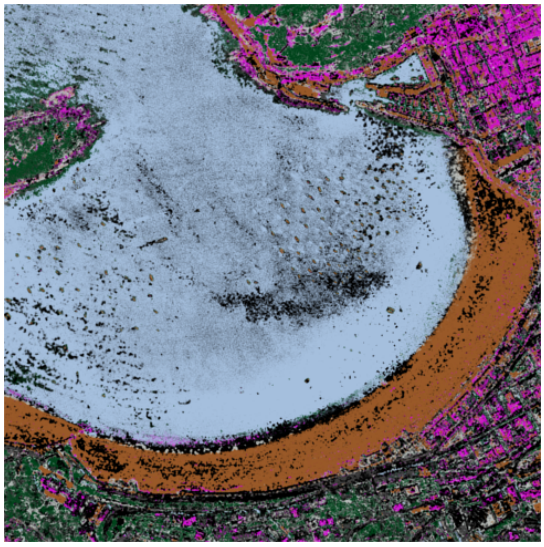
Mediante una mejor definición de las diferentes clases de cobertura, ampliando el entrenamiento, se pueden obtener mejores resultados. El mapa correspondiente obtenido, considerando todo el conjunto de descriptores incluyendo los descriptores de la Tabla 6.2, se muestra en la Figura 6.7c. Las medidas relativas a la calidad muestran resultados claramente comparables a los obtenidos por descriptores únicamente de color, incluso con mejoras en clases como *Playa*, que tiende a mezclarse con la clase colorimétricamente similar *Edificios*, pero geométricamente distinta.

Si trasladamos este test con todos los descriptores a las 5 áreas presentadas en 6.3, junto con el conjunto completo de 12 coberturas, obtenemos los resultados mostrados

(a)



(b)



(c)

Figura 6.7: Mapa de validación basado en mapas vectoriales preexistentes (a), resultados de clasificación supervisada con descriptores de color únicamente (b), así como con el conjunto completo de descriptores de la Tabla 6.2 en (c). Codificación de color por clase: *Edificios* (magenta), *Mar* (azul), *Bosques* (verde oscuro), *Área urbana* (gris), *Carreteras* (gris oscuro) y *Playa* (marrón). Una parte de los píxeles no han sido clasificados (negro).

| Estadísticas de rendimiento | Mar | | Bosque | | Zona urbana | |
|---|---|---|---|---|---|---|
| | Color | Full | Color | Full | Color | Full |
| Precisión : | 0.9861 | 0.9801 | 0.2438 | 0.2821 | 0.3294 | 0.3464 |
| Sensibilidad : | 0.9044 | 0.9241 | 0.4522 | 0.4247 | 0.2222 | 0.2494 |
| F1 : | 0.9435 | 0.9513 | 0.3168 | 0.3390 | 0.2654 | 0.2900 |
| Exactitud : | 0.9169 | 0.9299 | 0.8754 | 0.8929 | 0.8230 | 0.8535 |
| Estadísticas de rendimiento | Edificios | | Carreteras | | Playa | |
| | Color | Full | Color | Full | Color | Full |
| Precisión : | 0.3176 | 0.4419 | 0.1983 | 0.2643 | 0.4799 | 0.6418 |
| Sensibilidad : | 0.4358 | 0.3959 | 0.1856 | 0.2416 | 0.5723 | 0.9107 |
| F1 : | 0.3674 | 0.4176 | 0.1918 | 0.2524 | 0.5220 | 0.7529 |
| Exactitud : | 0.8629 | 0.9190 | 0.8179 | 0.8721 | 0.8460 | 0.9148 |

Tabla 6.4: Medidas de rendimiento en caso de clasificación con descriptores de color y con el conjunto completo de descriptores. Los resultados son comparables, con mejoras en el conjunto completo de descriptores en clases como playa, que tiende a confundirse con los tejados de la clase edificios, colorimétricamente similar, pero geométricamente distinta.

en las Figuras 6.8c, 6.9c, 6.10c, 6.11c y 6.12c. Para facilitar el análisis de los resultados obtenidos y comprobar *in situ* los problemas comentados previamente referentes al nivel de detalle y fechas de adquisición de los mapas de validación vectoriales, se ha creído oportuno realizar las composiciones mostradas en las Figuras 6.8, 6.9, 6.11, 6.10 y 6.12. Además de los resultados, con el propósito de constatar el nivel de resolución de la imágenes que son procesadas por el sistema o prototipo, en la Figura 6.13 se muestra una pequeña parte de una de las localizaciones en detalle correspondiente al recuadro rojo de la Figura 6.11a.

Las medidas de rendimiento del prototipo, con el conjunto de las 12 clases de cobertura evaluadas sobre todo el conjunto de localizaciones, se muestran en la Tabla 6.5.

Estadísticas de rendimiento

La clase con mejores resultados es *Mar*. Las clases *Arbustos* y *Campos* muestran un rendimiento más limitado; mientras que se han obtenido unos limitados resultados en las clases *Área industrial* y *Suelo desnudo*.

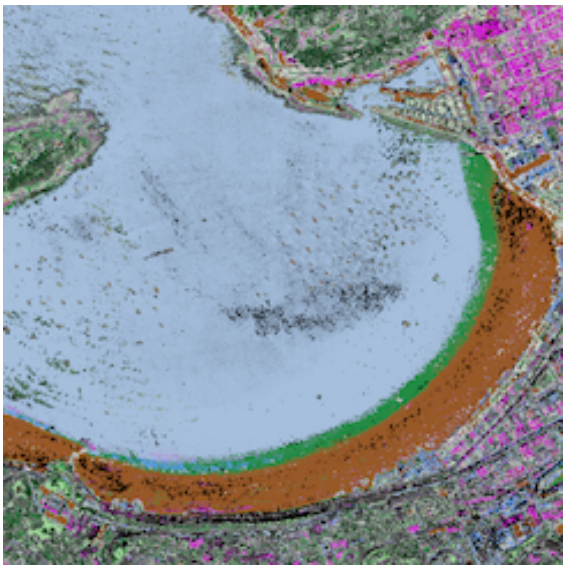
El análisis del mapa de validación vectorial muestra que los resultados están afectados significativamente por el escaso detalle en las zonas rurales. De ahí los buenos resultados de las clases *Arbusto* y *Campos*, como de una clase tan diversa como es el *Área industrial*. Otro efecto detectado se ha comentado anteriormente y está relacionado con la diferencia de fechas en los mapas vectoriales y la adquisición de la imagen, como se puede apreciar claramente en la zona costera sujeta a rápidos cambios como en el estuario de Urdaibai.

Esto apunta a la necesidad de una evaluación con respecto a un mapa de validación vectorial y a un mapa de validación detallado. Para completar la evaluación del rendimiento del sistema de clasificación del prototipo, se procede a evaluar los resultados obtenidos frente a un mapa de evaluación detallado.

Debido a los problemas ya comentados en la sección 6.1.4, el análisis sobre el mapa de evaluación detallado queda limitado a una de entre las 5 áreas localizadas: la bahía de "La Concha". La hemos elegido por contener el mayor número de clases diferentes -8 de entre las 12 categorías en 5 áreas disponibles-. Para que la diferencia en el número de clases no afecte a los resultados, se requiere reclasificar el área de la bahía sobre 8 clases y calcular los resultados frente a un mapa de validación de 8

(a)

(b)

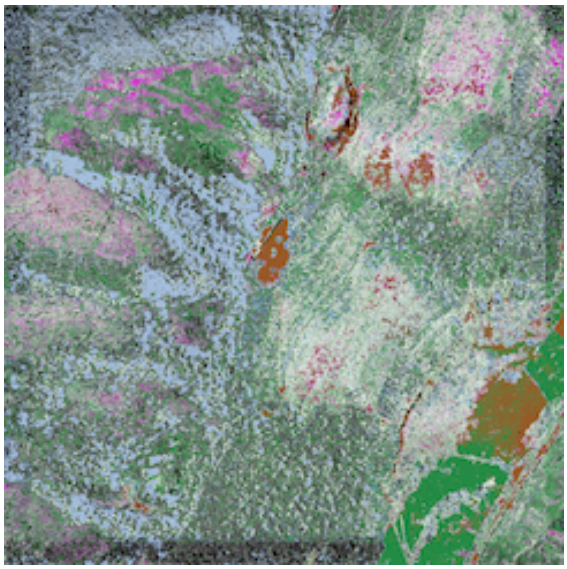


(c)

Figura 6.8: Composición resultados para bahía. a) Imagen original, b) Mapa de validación c) Resultado Clasificación.

(a)

(b)



(c)

Figura 6.9: Composición resultados para alta montaña. a) Imagen original, b) Mapa de validación c) Resultado Clasificación.

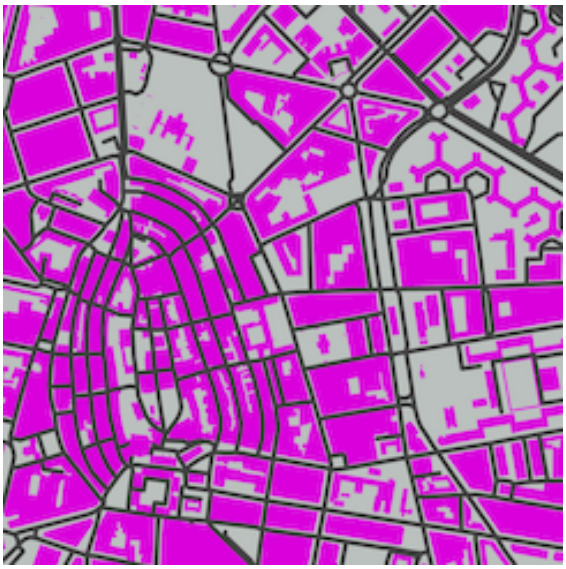
(a)

(b)

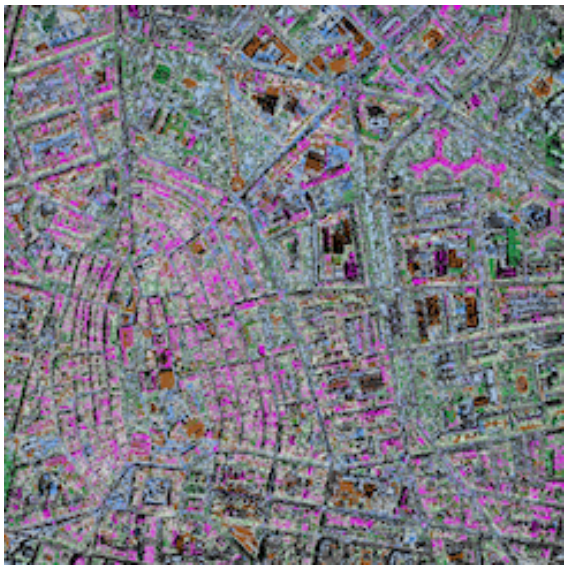
(c)

Figura 6.10: Composición resultados para zona industrial. a) Imagen original, b) Mapa de validación c) Resultado Clasificación.

(a)



(b)



(c)

Figura 6.11: Composición resultados para ciudad. a) Imagen original, b) Mapa de validación c) Resultado Clasificación.

(a)

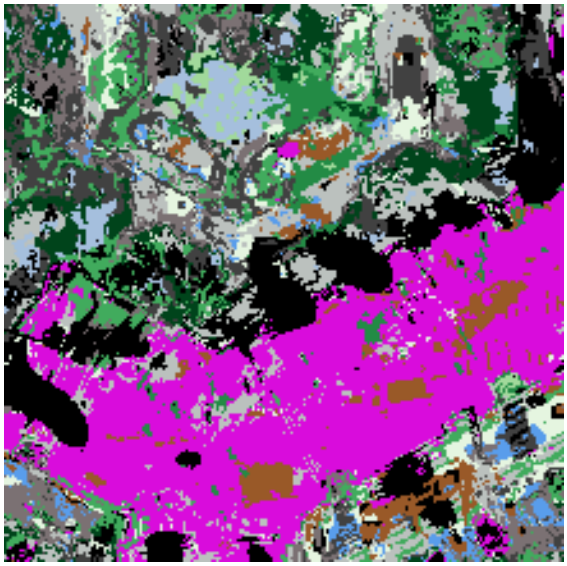
(b)

(c)

Figura 6.12: Composición resultados para reserva natural. a) Imagen original, b) Mapa de validación c) Resultado Clasificación.

(a)

(b)



(c)

Figura 6.13: Composición detalle 1:1 obtenida de imagen 6.11. a) Imagen original, b) Mapa de validación c) Resultado Clasificación.

| Estadísticas de rendimiento | Mar | Agua | Bosque | Zona urbana | Rocas | Pastos |
|---|---|---|---|---|---|---|
| Precisión : | 0.70 | 0.33 | 0.27 | 0.29 | 0.00 | 0.00 |
| Sensibilidad : | 0.90 | 0.13 | 0.13 | 0.18 | 0.00 | 0.00 |
| F1 : | 0.79 | 0.19 | 0.17 | 0.22 | 0.00 | 0.00 |
| Exactitud : | 0.83 | 0.70 | 0.58 | 0.57 | 0.83 | 0.86 |
| Estadísticas de rendimiento | Arbustos | Campos | Edificios | Zona industrial | Carreteras | Playa |
| Precisión : | 0.03 | 0.30 | 0.46 | 0.09 | 0.21 | 0.17 |
| Sensibilidad : | 0.29 | 0.39 | 0.17 | 0.13 | 0.24 | 0.77 |
| F1 : | 0.05 | 0.33 | 0.25 | 0.10 | 0.22 | 0.28 |
| Exactitud : | 0.80 | 0.68 | 0.66 | 0.72 | 0.61 | 0.74 |

Tabla 6.5: Medidas de rendimiento para el conjunto completo de 5 áreas basadas en mapas de validación vectoriales con 12 clases de cobertura. Los resultados de exactitud obtenidos son notablemente inferiores de los obtenidos considerando únicamente el área de La Concha debido al aumento de clases y la diferencia existente entre las imágenes originales y los mapas de validación generados.

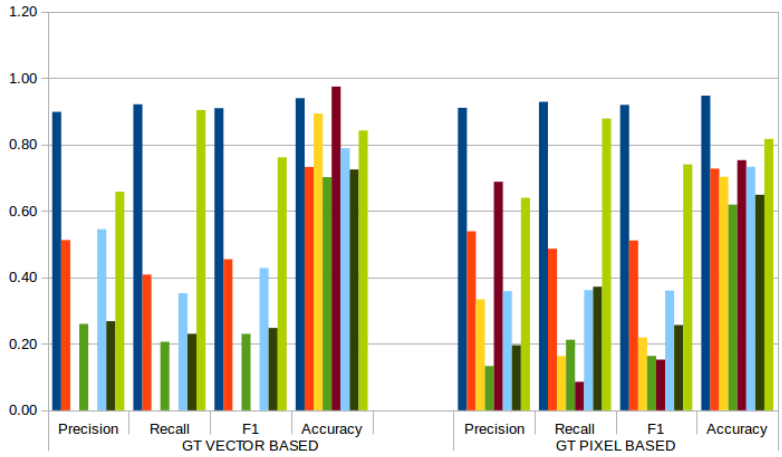
| Estadísticas de rendimiento | Mar | Bosque | Rocas | Zona urbana | Jardines | Edificios | Carretera | Playa |
|---|---|---|---|---|---|---|---|---|
| Precisión : | 0.99 | 0.28 | 0.00 | 0.37 | 0.00 | 0.52 | 0.03 | 0.69 |
| Sensibilidad : | 0.90 | 0.34 | 0.00 | 0.15 | 0.00 | 0.29 | 0.05 | 0.77 |
| F1 : | 0.94 | 0.31 | 0.00 | 0.22 | 0.00 | 0.37 | 0.04 | 0.73 |
| Exactitud : | 0.91 | 0.89 | 0.96 | 0.84 | 0.99 | 0.91 | 0.89 | 0.91 |

Tabla 6.6: Medidas de rendimiento sobre la clasificación sobre la bahía de La Concha respecto a mapas de validación vectoriales pre-existentes. La falta de las clases *Rocas* y *Jardines* en el mapa de validación no permite medir los parámetros estadísticos sobre su rendimiento

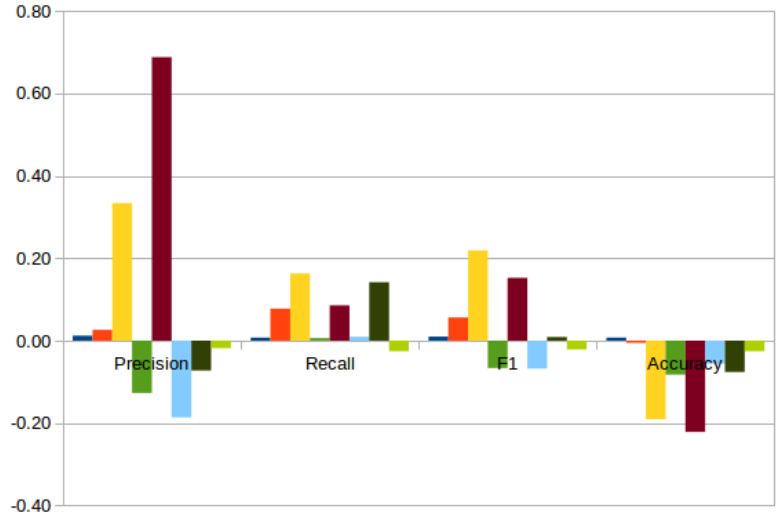
clases.

En la Tabla 6.6 se presentan los resultados para la evaluación, con el mapa de evaluación vectorial, mientras que en la Tabla 6.7 se presentan los resultados obtenidos con la evaluación del mapa detallado. La Figura 6.14 muestra una comparativa entre los dos resultados que facilita el análisis.

Las medidas de rendimiento obtenidas con el mapa de validación detallado muestran una mejora de entre el 20 % y 60 %, con respecto a los resultados obtenidos con el mapa de validación vectorial, debido a la disponibilidad de detalles no disponibles en los mapas y por la diferencia del momento de captura de la imagen y el mapa vectorial, donde la diferencia temporal supone un cambio importante, como en el estuario de Urdaibai. La clase *Jardines* es un ejemplo de ello, en donde los jardines y otros grupos de árboles dentro de las ciudades no están identificados en el mapa de validación vectorial. Las clases *Área Urbana* y *Carreteras* presentan un decrecimento en la precisión relacionado con el incremento de falsos positivos debido a una identificación diferente entre los dos mapas de validación, tal y como se puede apreciar en la esquina superior izquierda de las imágenes del centro y derecha de la Figura 6.6.




(a)




(b)

Figura 6.14: Medidas de calidad sobre la clasificación de las clases cobertura con respecto a dos tipos de mapas de validación (mapas vectoriales preexistentes en la izquierda y mapa de validación a nivel de píxel generado manualmente a la derecha) (a) y las diferencias entre los resultados obtenidos (b). Pares clase-color: Mar-Azul Marino, Bosque-Naranja, Rocas-Amarillo, Zona Urbana-Verde, Jardines-Marrón, Edificios-Azul Claro, Carreteras-Verde Oscuro y Playa-Verde Claro. Estos resultados se obtienen considerando únicamente el conjunto de datos de 'La Concha", donde el número de clases se ve reducido de 12 a 8. A pesar de que estos resultados no pueden extrapolarse directamente al conjunto prueba completo, indican que la falta de detalle espacial y la elección de una referencia temporal diferente con respecto a la adquisición de imagen pueden dar como resultado una diferencia de entre un 20 % y un 60 % en las medidas de rendimiento obtenidas. La clase *Jardines* es un ejemplo de ello, pues pequeños grupos de árboles no están identificados en el mapa de validación vectorial.

| Estadísticas de rendimiento | Mar | Bosque | Rocas | Zona urbana | Jardines | Edificios | Carretera | Playa |
|---|---|---|---|---|---|---|---|---|
| Precisión : | 0.99 | 0.48 | 0.06 | 0.22 | 0.55 | 0.66 | 0.16 | 0.73 |
| Sensibilidad : | 0.87 | 0.48 | 0.08 | 0.14 | 0.07 | 0.30 | 0.30 | 0.92 |
| F1 : | 0.93 | 0.48 | 0.07 | 0.17 | 0.13 | 0.41 | 0.21 | 0.81 |
| Exactitud : | 0.90 | 0.91 | 0.95 | 0.87 | 0.94 | 0.91 | 0.92 | 0.94 |

Tabla 6.7: Medidas de rendimiento sobre la clasificación sobre la bahía de La Concha respecto a mapas de validación basado en interpretación manual a nivel de píxel. Estos resultados se obtienen considerando únicamente el entorno de validación de La Concha, donde el número de clases se ve reducido desde las 12 originales a 8.

## 6.4. Resultados sobre el rendimiento de indexación

Como se comenta en la Sección 6.2.2, la evaluación ha consistido en la comparación de tres motores de indexación con diferentes grados de optimización. En este sentido, los valores obtenidos con los diferentes motores de indexación se muestran en la Figura 6.15.

Como se puede apreciar en la Figura 6.15, la diferencia entre los resultados obtenidos por las diferentes medidas de rendimiento de la clasificación, en los tres motores no parece ser destacable más que en algún caso como la sensibilidad en las clases *Vegetación* y *Edificios*.

Para la medición del tiempo de procesamiento se han considerado las etapas de clasificación que incluyen el entrenamiento, la evaluación, y fusionado. Esta última etapa es la más costosa del proceso, puesto que no solo depende del número de clases en las que se ha podido clasificar cada píxel sino que, además, al buscar el elemento de entrenamiento más cercano, es dependiente del volumen de entrenamiento.

Los resultados obtenidos en tiempo de procesamiento están reflejados en la Tabla 6.8.

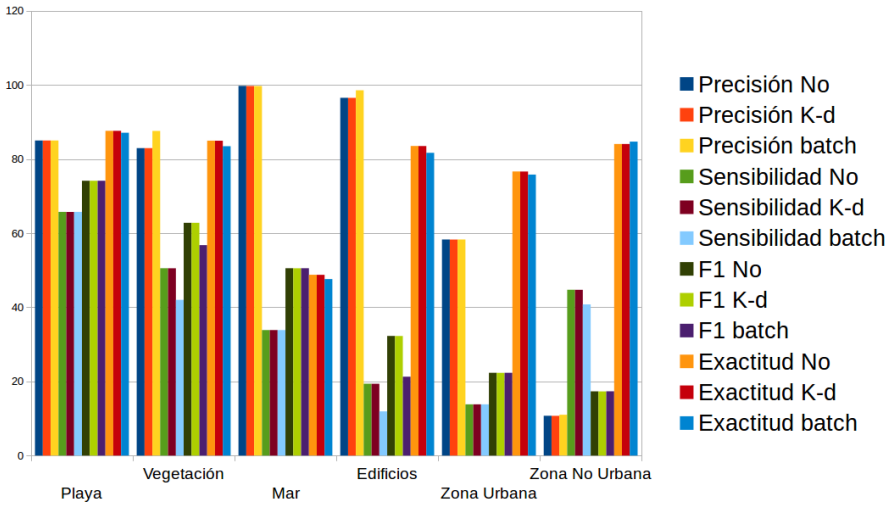



Figura 6.15: Comparativa de resultados estadísticos entre diferentes motores de indexación. Los valores de rendimiento similares obtenidos habilitan la comparación sobre el tiempo de procesamiento.

| Optimización algorítmica sobre $k$-Vecinos más cercanos | Tiempo de procesamiento (HH:mm:ss,00) |
|---|---|
| Ninguna | 6:00:35.91 |
| Árbol $k$-d con consultas puntuales | 00:25:43.75 |
| Árbol $k$-d con consulta conjuntas | 00:00:56.89 |

Tabla 6.8: Comparativa de motores de indexación. La obtención de valores similares en el rendimiento de la clasificación permite la comparativa del tiempo de procesamiento con resultados muy significativa entre los diferencia motores de indexación.

Tal y como se aprecia, la comparación entre los resultados obtenidos es muy significativa. La reducción del tiempo entre el motor de indexación no optimizado y el optimizado con consultas puntuales supone el 7 % del primero, mientras que el tiempo entre el optimizado con consultas puntuales y el optimizado con consultas conjuntas supone el 3 % del primero y un $0,02$ %, si lo comparamos con el motor de indexación sin optimizar.

En cuanto al rendimiento de indexación se ha considerado oportuno analizar el efecto causado por diferentes abstracciones *RDD*. Para lo cual, se ha variado el tamaño de arista de la cuadrícula en las que se dividen los mapas.

Para ello se ha desarrollado la siguiente evaluación: por un lado, además del área de evaluación, centrada en la bahía de La Concha de Donostia – San Sebastián, se ha adjuntado otro set de evaluación, ampliando el área de análisis a toda la extensión del País Vasco, para considerar un volumen de datos más cercano al paradigma *Big Data* del que se tiene con solo la bahía. Por otro lado, se han considerado 4 tamaños diferentes de aristas sobre las que evaluar el rendimiento de la arquitectura propuesta. Los datos obtenidos se muestran en las Tablas 6.9 y 6.10.

Tanto en una tabla como en otra, la diferencia de tiempo de procesamiento es considerable, apreciándose claramente el efecto generado por un tamaño de arista frente a otro. De este modo, se certifica la importancia y relevancia que adquiere la elección del tamaño del bloque de datos a distribuir en el tiempo de procesamiento necesario.

| Localización procesamiento | Tamaño cuadrícula píxeles | Tiempo procesamiento (segundos) |
|---|---|---|
| Proceso mono CPU | 256 | 56.89 |
| Spark cluster 4 CPU | 64 | 320.07 |
| Spark cluster 4 CPU | 128 | 21.47 |
| Spark cluster 4 CPU | 256 | 39.14 |
| Spark cluster 4 CPU | 512 | 62.68 |

Tabla 6.9: Tiempos de procesamiento del área de Donostia-San Sebastián en función del tamaño de la cuadrícula en local y en cluster para la computación distribuida. Sistema Intel® Core™ i7-4750HQ (2.0GHZ), 8 GB de memoria RAM y disco duro SSD.

| Localización procesamiento | Tamaño cuadrícula píxeles | Tiempo procesamiento (horas) |
|---|---|---|
| Proceso mono CPU | 256 | 21.93 |
| Spark cluster 4 CPU | 64 | 124.47 |
| Spark cluster 4 CPU | 128 | 8.34 |
| Spark cluster 4 CPU | 256 | 15.22 |
| Spark cluster 4 CPU | 512 | 24.37 |

Tabla 6.10: Tiempos de procesamiento del área País Vasco en función del tamaño de la cuadrícula en local y en cluster para la computación distribuida. Sistema Intel® Core™ i7-4750HQ (2.0GHZ), 8 GB de memoria RAM y disco duro SSD.

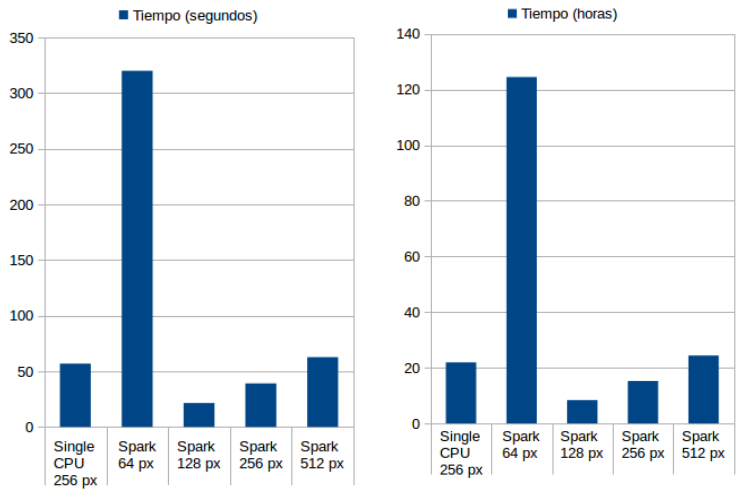



Figura 6.16: Comparativa de tiempo de ejecución en local y en la nube. Visualización de los datos presentado en las Tablas 6.9 y 6.10 para una mejor interpretación de los resultados. En ambos casos, es significativa la relevancia del tamaño de la arista de la cuadrícula, para una óptima y eficiente distribución del trabajos y del procesamiento.

# Capítulo 7

# Conclusiones y líneas futuras

En el proceso doctoral que ha dado como resultado la presente tesis, se ha comprobado la viabilidad y el valor aplicativo de la integración de funcionalidades de aprendizaje automático en servidores mapas web para la generación de mapas temáticos personalizados.

La validación de la hipótesis presentada se ha realizado mediante la implementación y evaluación de un prototipo pre-operacional para el mapeo de datos temáticos sobre imágenes de teledetección de muy alta resolución mediante aprendizaje supervisado a través de una plataforma web. Integrando las capacidades de escalabilidad del aprendizaje automático y de los servidores de mapas web, la hipótesis supera el estado del arte actual, caracterizado por la separación de los dos ámbitos que requieren la continua intervención del experto de teledetección en tareas de mapeo temático intensivo. El esquema conceptual de la metodología desarrollada en el sistema implementado se presenta en la Figura 7.1.

Frente a un estado del arte con ejemplos de sistemas que integran web mapping y minería de datos no supervisados, se ha diseñado e implementado un sistema preoperacional basado en un algoritmo típicamente utilizado en bases de datos, como es el *T-k-PNN* [24], adaptándolo a tareas de minería de datos y, más específicamente, a la clasificación supervisada sobre imágenes de teledetección, optimizado mediante árboles $k$-d [108].

La optimización del algoritmo se ha validado mediante la evaluación de los tiempos de procesamiento requeridos por el prototipo, utilizando diferentes motores de

indexación. Una vez evaluado el motor de indexación, se ha evaluado el tamaño óptimo para la distribución de los datos y su efecto en el tiempo de procesamiento [109].

Debido a los volúmenes de datos empleados típicamente en el ámbito de la teledetección, la implementación de los algoritmos ha sido trasladada a la nube de acuerdo con los paradigmas de *Big Data*, evaluándose el rendimiento del sistema [109].

El aprendizaje automático se ha integrado en un entorno web, obteniendo un funcionamiento satisfactorio del sistema en términos de tiempo de respuesta y rendimiento [110].. La validación del rendimiento del algoritmo propuesto mediante análisis estadístico se ha llevado a cabo gracias a la utilización de mapas de validación desarrollados para ello. Se han desarrollado dos tipos de mapas de acuerdo a las necesidades encontradas en el proceso de validación [108].

Se ha configurado una interfaz de usuario que permite personalizar el aprendizaje automático -adaptándola a cada caso de uso- y el análisis visual de los resultados, permitiendo al usuario optimizar los resultados obtenidos [106].

Finalmente, el trabajo de tesis presentado se ha completado presentando un segundo prototipo con el que se han querido integrar las tareas de teledetección con el paradigma de Analítica Visual (apéndice A) [105].

Todo este trabajo se ha visto reflejado en las diferentes contribuciones científicas realizadas en el ámbito de la cartografía y la teledetección, presentadas en diferentes congresos y revistas científicas [105, 106, 108–112].

El presente trabajo de tesis doctoral deja abiertas lineas de investigación y nuevos trabajos en los diferentes ámbitos estudiados.

Relacionado con la clasificación supervisada, el desarrollo o implementación de algoritmos que mejoren los resultados estadísticos obtenidos mejorando el rendimiento en términos de tiempo de procesamiento es una tarea de continua mejora. La inclusión de imágenes hiperespectrales enriquecerá los datos actuales, abriendo un nuevo campo de caracterización que habrá que tratar. La implementación de un prototipo completamente funcional en la nube que siga los estándares descritos por el *Open Geospatial Consortium* es otra tarea a completar.

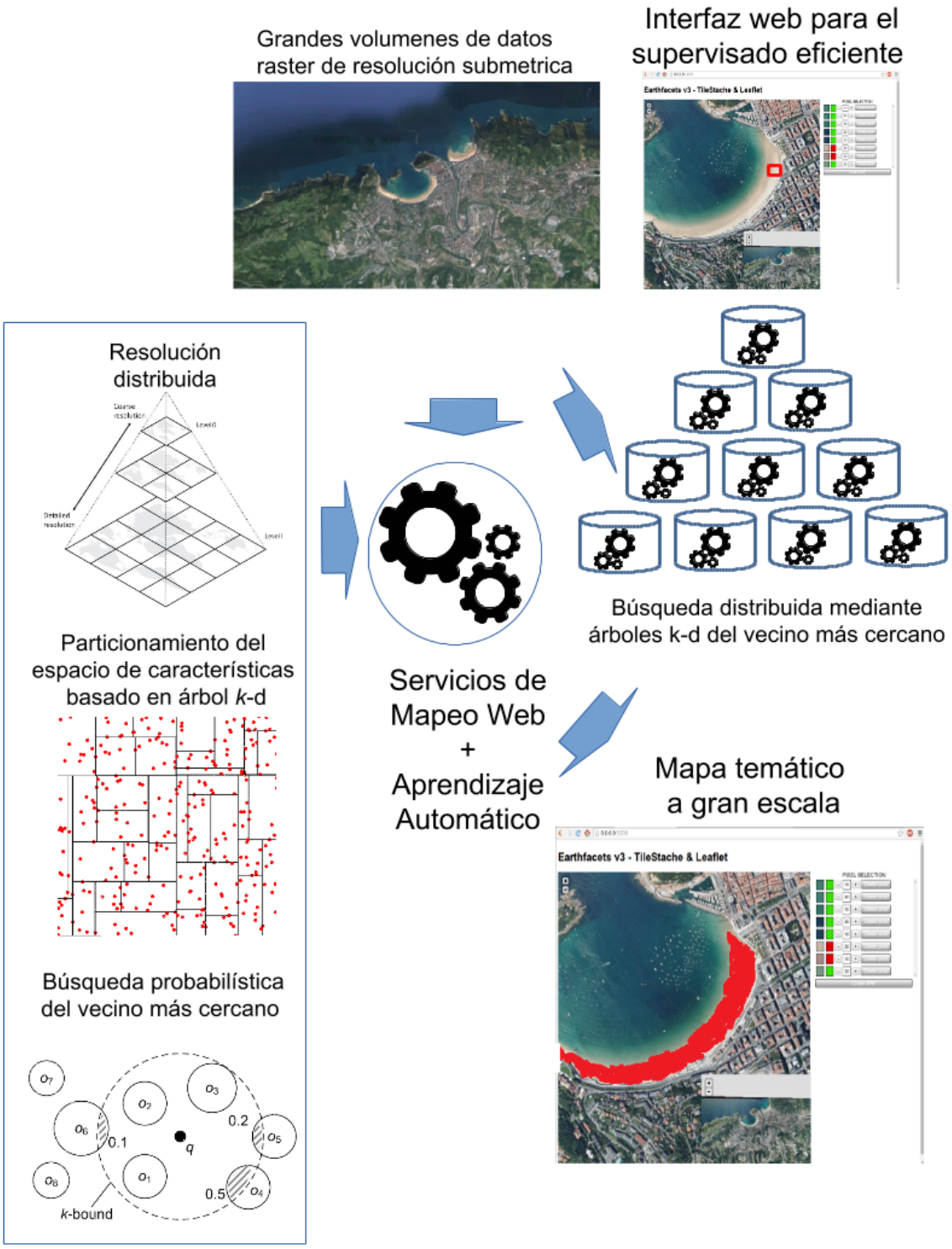



Figura 7.1: Esquema conceptual de la metodológica desarrollada en el sistema implementado. El sistema se compone de un motor escalable capaz de operar en entornos distribuidos para el mapeo vía web para la generación de mapas temáticos personalizados desde datos de teledetección de muy alta resolución

# Apéndice A

# Analítica visual en el entorno de teledetección

Una manera de facilitar la tarea de selección de características, puede ser la incorporación del usuario en el análisis para la selección interactiva de características de interés. En un primer acercamiento considerado [113] utiliza un visor tridimensional en el que se visualizan las cuadrículas como se puede ver en la figura A.1. La tarea de selección en un visor 3D dificulta la selección de las imágenes de interés. Incorporar al usuario requiere desarrollar una interfaz interactiva 2D, donde se facilite el proceso de recolección de la información. Desde el punto de vista de la teledetección, explotar el conocimiento de usuarios no expertos en este ámbito requiere el diseño de aplicaciones de fácil uso. Por ello, el resto del apéndice presenta un prototipo que aúna la teledetección con técnicas con técnicas de analítica visual.

Finalmente se presenta un prototipo que aúna la teledetección con técnicas de analítica visual A. Las analíticas visuales muestran detalles de su implementación en la sección A.2. presentación de varios ejemplos de uso de la combinación de técnicas de analítica visual en teledetección, en la sección A.3.

## A.1. Enfoque metodológico

Las metodologías de Analítica Visual, en inglés *Visual Analytics*[114] para la caracterización del contenido pueden ser implementadas vía web, mediante una interfaz

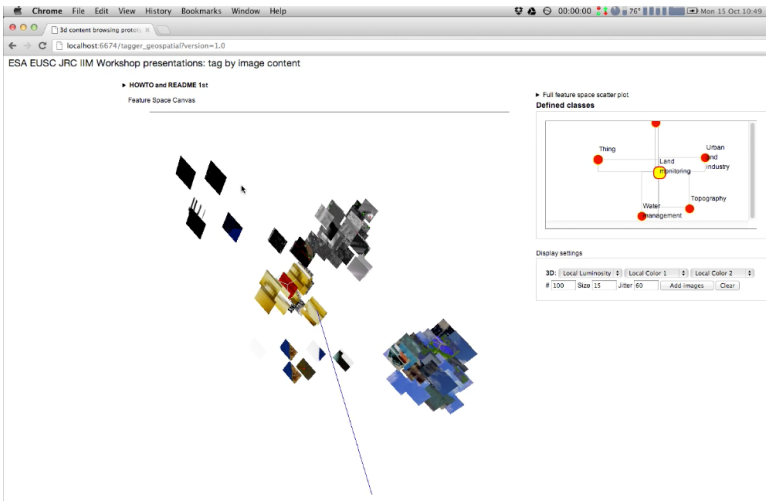



Figura A.1: Interfaz de usuario basada en galería 3D de imagen de cuadrícula: la interacción del usuario para la selección de elementos de interés requiere habilidades avanzadas de navegación en un alto espacio dimensional [113].

que integre al usuario en el bucle de interpretación y le permita filtrar características basadas en contenido para obtener el conjunto de imágenes objetivo.

Aplicar razonamiento analítico mediante representaciones visuales implica aspectos metodológicos relacionados tanto con el diseño de múltiples visualizaciones interactivas como consideraciones en las representaciones y las transformaciones de datos. A continuación, se presentan ejemplos de dichos aspectos metodológicos destinados a la comprensión y caracterización de conjuntos de datos de resolución métrica adquiridas en entornos urbanos.

Este enfoque supone el desarrollo de una interfaz basada en una galería de imágenes, además de la introducción de herramientas de analítica visual que faciliten la selección de imágenes de interés. En esta primera implementación de herramientas para el análisis visual, se han implementado gráficos de barras para visualizar los datos extraídos de las imágenes de la colección, ver Figura A.3.

Del mismo modo que las capacidades de procesamiento sobre volúmenes de datos a gran escala están limitadas en los equipos de sobremesa actuales, las capacidades

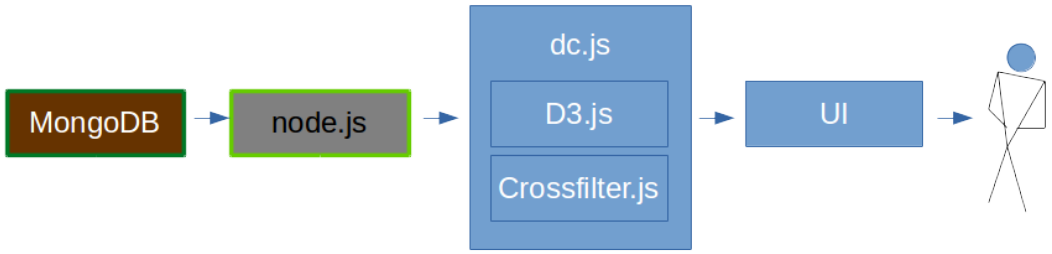



Figura A.2: Arquitectura de alto nivel del sistema. Los metadatos correspondientes a las imágenes de la cuadrícula son almacenados en una base de datos noSQL a través de una interfaz completamente REST asíncrona.

de carga de datos en una interfaz web también se ven limitadas. Para hacer frente a este problema, se ha implementado un sistema de consultas basado en el paradigma *MapReduce* en MongoDB que soporta la programación del modelo *MapReduce* en el lado servidor.

## A.2. Integración de analíticas visuales en teledetección

La arquitectura del sistema está basada en un modelo de tres capas. Los datos están almacenados en una base de datos no-SQL como MongoDB[115] capaz de un reparto y ejecución de tareas *MapReduce*. La comunicación entre la interfaz de usuario en la parte del cliente y la base de datos se realiza a través de eventos asíncronos en un servidor de aplicaciones web altamente paralelizables.

El servidor implementa una API HTTP completamente *RESTful* [116]. La interfaz de usuario realiza consultas al servidor a través de llamadas HTTP, representando internamente los objetos recuperados en un formato orientado hacia el modelo OLAP de manera eficiente [117].

Como se puede ver en la Figura A.3, la interfaz de usuario dispone de dos partes bien diferenciadas, un panel de resultados o galería de imágenes y un panel de control o filtraje. La galería muestra un conjunto de imágenes aleatorias correspondientes a las cuadrículas en las que se divide típicamente un mapa. Al mismo tiempo que el usuario selecciona diferentes rangos en los descriptores está filtrando los elementos

que disponemos en la base datos, de manera que la galería actualiza las imágenes que debe mostrar según se realizan las operaciones de selección.

Una de las características más interesantes de la interfaz es el funcionamiento interactivo y coordinado que poseen todas las visualizaciones, tanto la galería como los descriptores de características. Cuando el usuario selecciona o filtra un rango en los descriptores de los datos, la galería y el resto de descriptores actualizan la información a visualizar en tiempo real.

## A.3. Ejemplo de uso de analítica visual en teledetección

Debido a las caracteristicas de los datos sólo se trata la información relativa a espacio de color estándar RGB y a su transformado en el espacio HSV.

Como ya se ha comentado, el análisis visual puede facilitar el análisis de enormes volúmenes de datos. En esta implementación los datos provienen de imágenes DigitalGlobe del centro de Roma, con una resolución de píxel por metro cuadrado. El tamaño de la arista para la cuadrícula es de 100x100 píxeles, siendo completamente arbitraria para este primer test. Si se realiza un rápido análisis visual sobre las imágenes, se pueden detectar diferentes elementos como edificios nuevos e históricos, diferentes tipos de vegetación como árboles y jardines, un río y carreteras, entre otros elementos. Estos elementos pueden considerarse como clases, dentro de una tarea de clasificación supervisada.

Cuando se accede a la página web, en la interfaz de usuario se cargan al mismo tiempo todos los datos necesarios para la visualización. Una vez los datos están cargados, la acción de selección y por consiguiente filtrado no requiere más comunicación con el servidor haciendo la interacción más dinámica y con mayor capacidad de respuesta.

En las siguientes figuras se muestran los resultados de los experimentos realizados. En la Figura A.4, la selección sobre los diferentes histogramas da como resultado imágenes del río Tiber. Las cuadrículas son identificadas como zonas muy luminosas, con una saturación significativa y un fuerte componente verde. Cabe destacar que la selección de características genera la actualización tanto en la galería de imágenes

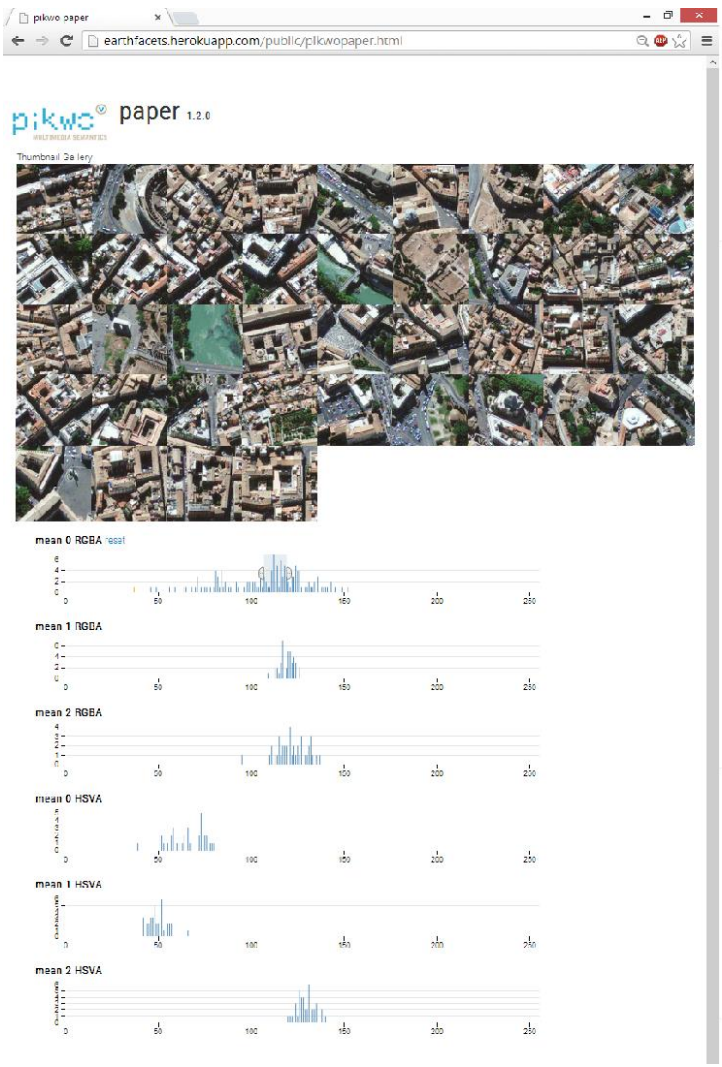



Figura A.3: La galería de imágenes en la parte superior muestra una selección aleatoria de las cuadrículas disponibles. El usuario puede interactuar con la galería con la parte inferior de la interfaz. En esta sección se presentan los diferentes histogramas correspondientes a cada una de las características extraídas de todo el conjunto de datos. Mediante la selección de un rango de valores en estos histogramas el usuario selecciona una región de interés correspondiente al espacio de características. La selección puede ser redefinida y optimizada moviendo los límites o los centros de las selecciones. Cada selección genera una actualización en tiempo real, tanto de la galería como de los histogramas colindantes. Se permiten múltiples selecciones concurrentes para definir el hipercubo seleccionado en el espacio de características.

como en el resto de los histogramas.

En la siguiente imagen A.5, la selección da como resultado cuadrículas con árboles. Las áreas arboladas se identifican por zonas de baja luminosidad y con alta saturación, debido a las sombras visibles en ellas. La selección de un segmento específico en el matiz del color, parámetro H en el espacio de color HSV, hace que sea innecesario el uso de la componente verde, para el filtrado en el espacio de color RGB.

Finalmente en la Figura A.6 se han podido obtener imágenes con edificios históricos. La selección de intervalos específicos, el matiz y la luminosidad son suficientes para la selección de esta clase de imágenes muy presentes en el conjunto de datos disponible.

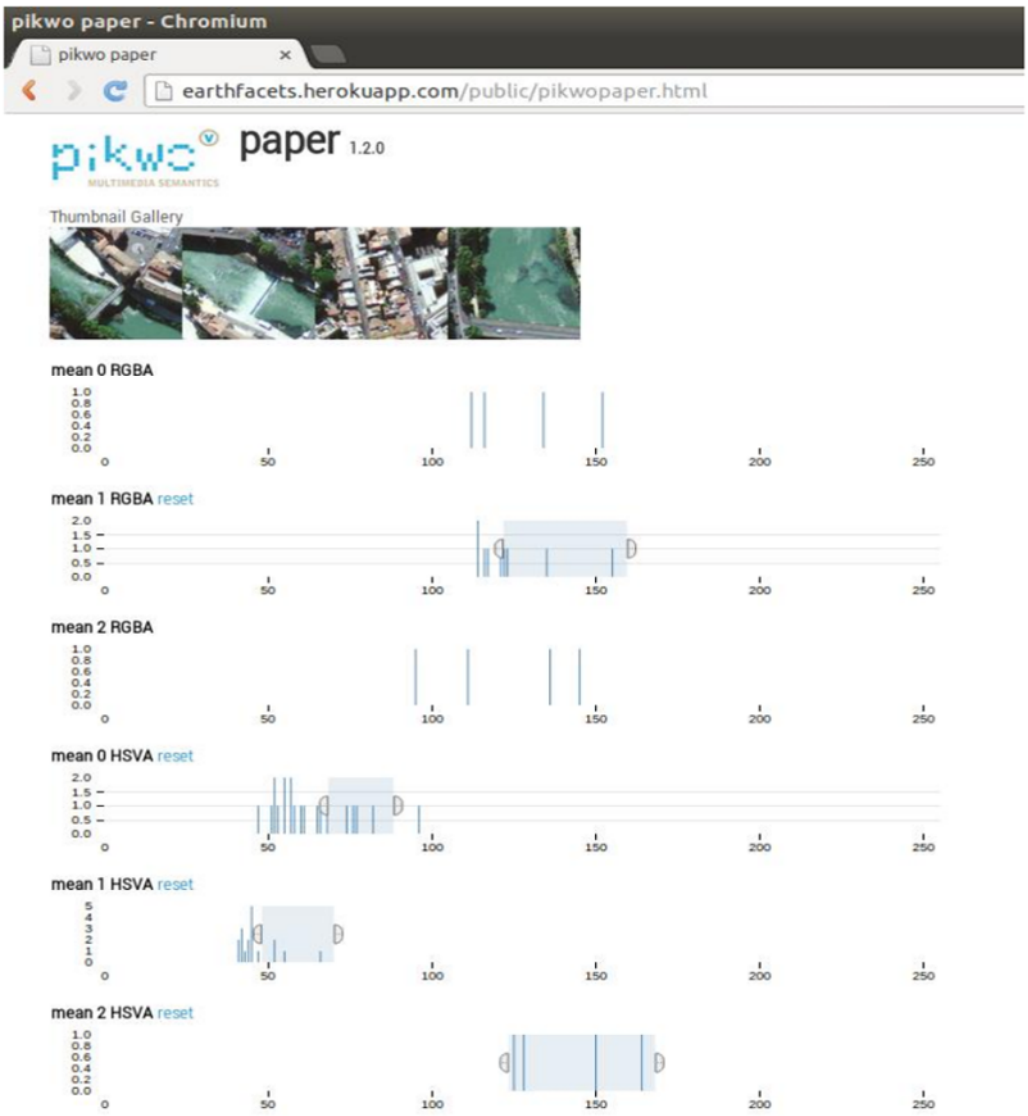



Figura A.4: Resultados de la búsqueda de ríos: Las cuadrículas son identificadas como zonas muy luminosas, con una saturación significativa y un fuerte componente verde. Destaca que la selección de características ha producido la actualización tanto en la galería de imágenes, como en el resto de los histogramas.


Figura A.5: Resultados de la búsqueda de árboles: Las áreas arboladas se identifican por zonas de baja luminosidad y con alta saturación, debido a las sombras visibles en ellos. La selección de un segmento específico en el matiz del color, parámetro H en el espacio de color HSV, hace que sea innecesario el uso de la componente verde para el filtrado en el espacio de color RGB.


Figura A.6: Resultados de la búsqueda de edificios históricos: la selección de intervalos específicos el matiz y la luminosidad son suficientes para la selección de esta clase de imágenes muy presentes en el conjunto de datos disponible.

# Bibliografía


[1] M. Quartulli and I. G. Olaizola, "A review of {EO} image information mining," {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, vol. 75, no. 0, pp. 11 – 28, 2013.

[2] K. Evangelidis, K. Ntouros, S. Makridis, and C. Papatheodorou, "Geospatial services in the cloud," *Computers & Geosciences*, vol. 63, no. 0, pp. 116 – 122, 2014.

[3] C. Lee, S. Gasster, A. Plaza, C.-I. Chang, and B. Huang, "Recent developments in high performance computing for remote sensing: A review," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE journal of*, vol. 4, no. 3, pp. 508–527, 2011.

[4] C. E. Woodcock, *Uncertainty in Remote Sensing.* John Wiley & Sons, Ltd, 2006, pp. 19–24.

[5] S. Krug, *Don'T Make Me Think: A Common Sense Approach to the Web (2Nd Edition).* Thousand Oaks, CA, USA: New Riders Publishing, 2005.

[6] M. D. Hill, "What is scalability?" *SIGARCH Comput. Archit. News*, vol. 18, no. 4, pp. 18–21, 1990.

[7] R. Gitzel, A. Korthaus, and M. Schader, "Using established web engineering knowledge in model-driven approaches," *Science of Computer Programming*, vol. 66, no. 2, pp. 105 – 124, 2007.

[8] F. F.-H. Nah, "A study on tolerable waiting time: how long are web users

willing to wait?" *BEHAVIOUR & INFORMATION TECHNOLOGY*, vol. 23, no. 3, pp. 153–163, 2004.

[9] T. Lim, W. Loh, and Y. Shih, "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *MACHINE LEARNING*, vol. 40, no. 3, pp. 203–228, 2000.

[10] M. Ali-ud-din Khan, M. Uddin, and N. Gupta, "Seven v's of big data understanding big data to extract value," in *American Society for Engineering Education (ASEE Zone 1), 2014 Zone 1 Conference of the*, April 2014, pp. 1–5.

[11] D. Keim, H. Qu, and K.-L. Ma, "Big-data visualization," *IEEE Computer Graphics and Applications*, vol. 33, no. 4, pp. 20–21, 2013.

[12] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds., *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996.

[13] I. Guyon, "A scaling law for the validation-set training-set size ratio," in *AT & T Bell Laboratories*, 1997.

[14] C. Ordoñez and E. Omiecinski, "Image mining: a new approach for data mining," Georgia Institute of Technology, Tech. Rep., 1998.

[15] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.

[16] H. Parvin, H. Alizadeh, and B. Minaei-Bidgoli, "Mknn: Modified k-nearest neighbor," in *Proceedings of The World Congress on Engineering and Computer Science 2008*, International Association of Engineers. Newswood Limited, 2008, pp. 831–834.

[17] Q. He and J. Wang, "Principal component based k-nearest-neighbor rule for semiconductor process fault detection," in *American Control Conference, 2008*, 2008, pp. 1606–1611.

[18] D. Barbará, H. Garcia-Molina, and D. Porter, "The management of probabilistic data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 4, no. 5, pp. 487–502, 1992.

[19] R. Cheng, D. V. Kalashnikov, and S. Prabhakar, "Evaluating probabilistic queries over imprecise data," in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 2003, pp. 551–562.

[20] N. Dalvi and D. Suciu, "Efficient query evaluation on probabilistic databases," *The VLDB journal*, vol. 16, no. 4, pp. 523–544, 2007.

[21] P. Agrawal, O. Benjelloun, A. D. Sarma, C. Hayworth, S. Nabar, T. Sugihara, and J. Widom, "Trio: A system for data, uncertainty, and lineage," in *Proceedings of the 32Nd International Conference on Very Large Data Bases*. VLDB Endowment, 2006, pp. 1151–1154.

[22] A. P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao, "Querying the uncertain position of moving objects." in *Temporal Databases, Dagstuhl*, 1997, pp. 310–337.

[23] D. Pfoser and C. Jensen, "Capturing the uncertainty of moving-object representations," in *Advances in Spatial Databases*, R. Güting, D. Papadias, and F. Lochovsky, Eds. Springer Berlin Heidelberg, 1999, vol. 1651, pp. 111–131.

[24] R. Cheng, L. Chen, J. Chen, and X. Xie, "Evaluating probability threshold k-nearest-neighbor queries over uncertain data," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM, 2009, pp. 672–683.

[25] G. S. Iwerks, H. Samet, and K. Smith, "Continuous k-nearest neighbor queries for continuously moving points with updates," in *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29*. VLDB Endowment, 2003, pp. 512–523.

[26] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*. VLDB Endowment, 2004, pp. 588–599.

[27] S. Ganguly, M. N. Garofalakis, R. Rastogi, and K. K. Sabnani, "Streaming algorithms for robust, real-time detection of ddos attacks," in *27th IEEE International Conference on Distributed Computing Systems (ICDCS 2007), June 25-29, 2007, Toronto, Ontario, Canada*, 2007, p. 4.

[28] J. Walters-Williams and Y. Li, "Comparative study of distance functions for nearest neighbors," in *Advanced Techniques in Computing Sciences and Software Engineering*, 2010, pp. 79–84.

[29] T. Waheed, R. B. Bonnell, S. O. Prasher, and E. Paulet, "Measuring performance in precision agriculture: CART - A decision tree approach," *AGRICULTURAL WATER MANAGEMENT*, vol. 84, no. 1-2, pp. 173–185, 2006.

[30] Z. Wang, L. Xue, and D. Feng, "Mining textural association rules in RS image," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 6790, 2007, pp. 67 902J–1–7.

[31] J. Vauhkonen and L. Mehtatalo, "Matching remotely sensed and field-measured tree size distributions," *CANADIAN journal OF FOREST RESEARCH*, vol. 45, no. 3, pp. 353–363, 2015.

[32] S. Ghaffarian and S. Ghaffarian, "Automatic histogram-based fuzzy c-means clustering for remote sensing imagery," {*ISPRS*} *journal of Photogrammetry and Remote Sensing*, vol. 97, no. 0, pp. 46 – 57, 2014.

[33] M. Schröder, H. Rehrauer, K. Seidel, and M. Datcu, "Interactive learning and probabilistic retrieval in remote sensing image archives," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 38, pp. 2288–2298, 2000.

[34] H. Costa, H. C. no, F. Baçao, and M. Caetano, "Combining per-pixel and object-based classifications for mapping land cover over large areas," *Int. J. Remote Sens.*, vol. 35, no. 2, pp. 738–753, 2014.

[35] G. P. Petropoulos, D. P. Kalivas, I. A. Georgopoulou, and P. K. Srivastava, "Urban vegetation cover extraction from hyperspectral imagery and geographic information system spatial analysis techniques: case of athens, greece," *journal of Applied Remote Sensing*, vol. 9, no. 1, p. 096088, 2015.

[36] S. W. Myint, P. Gober, A. Brazel, S. Grossman-Clarke, and Q. Weng, "Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery," *Remote Sensing of Environment*, vol. 115, no. 5, pp. 1145 – 1161, 2011.

[37] D. Dubois and R. Lepage, "Fast and efficient evaluation of building damage from very high resolution optical satellite images," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE journal of*, vol. 7, no. 10, pp. 4167–4176, 2014.

[38] U. Maulik and A. Sarkar, "Efficient parallel algorithm for pixel classification in remote sensing imagery," *GeoInformatica*, vol. 16, no. 2, pp. 391–407, 2012.

[39] Q. Ho, P. Lundblad, T. Aström, and M. Jern, "A web-enabled visualization toolkit for geovisual analytics," *Information Visualization*, vol. 11, no. 1, pp. 22–42, 2012.

[40] P. E. Keel, "Collaborative visual analytics: Inferring from the spatial organization and collaborative use of information," in *2006 IEEE Symposium On Visual Analytics Science And Technology*. IEEE, 2006, pp. 137–144.

[41] P. Cappelaere, S. Sanchez, S. Bernabe, A. Scuri, D. Mandl, and A. Plaza, "Cloud implementation of a full hyperspectral unmixing chain within the nasa web coverage processing service for eo-1," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE journal of*, vol. 6, no. 2, pp. 408–418, 2013.

[42] M. J. Cracknell and A. M. Reading, "Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit

spatial information ," *Computers & Geosciences*, vol. 63, no. 0, pp. 22 – 33, 2014.

[43] W. Lee, V. Alchanatis, C. Yang, M. Hirafuji, D. Moshou, and C. Li, "Sensing technologies for precision specialty crop production," *Computers and Electronics in Agriculture*, vol. 74, no. 1, pp. 2–33, 2010.

[44] C. R. Medlin, D. R. Shaw, P. D. Gerard, and F. E. LaMastus, "Using remote sensing to detect weed infestations in glycine max," *Weed Science*, vol. 48, no. 3, pp. pp. 393–398, 2000.

[45] M. Simard, S. Saatchi, and G. De Grandi, "The use of decision tree and multiscale texture for classification of jers-1 sar data over tropical forest," *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, vol. 38, no. 5, 1, pp. 2310–2321, 2000.

[46] D. Goodenough, A. Dyk, O. Niemann, J. Pearlman, H. Chen, T. Han, M. Murdoch, and C. West, "Processing hyperion and ali for forest classification," *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, vol. 41, no. 6, 1, pp. 1321–1331, 2003.

[47] G. Carpenter, M. Gjaja, S. Gopal, and C. Woodcock, "Art neural networks for remote sensing: Vegetation classification from landsat tm and terrain data," *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, vol. 35, no. 2, pp. 308–325, 1997.

[48] S. Quegan, T. Le Toan, J. Yu, F. Ribbes, and N. Floury, "Multitemporal ers sar analysis applied to forest mapping," *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, vol. 38, no. 2, 1, pp. 741–753, 2000.

[49] K. Ranson, S. Saatchi, and G. Sun, "Boreal forest ecosystem characterization with SIR-C/XSAR," *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, vol. 33, no. 4, pp. 867–876, 1995.

[50] A. Pacheco, J. Horta, C. Loureiro, and O. Ferreira, "Retrieval of nearshore bathymetry from landsat 8 images: A tool for coastal monitoring in shallow

waters," *REMOTE SENSING OF ENVIRONMENT*, vol. 159, pp. 102–116, 2015.

[51] A. Suo and M. Zhang, "Sea areas reclamation and coastline change monitoring by remote sensing in coastal zone of liaoning in china," *journal OF COASTAL RESEARCH*, no. 73, pp. 725–729, 2015.

[52] A. M. Dewan and Y. Yamaguchi, "Using remote sensing and gis to detect and monitor land use and land cover change in dhaka metropolitan of bangladesh during 1960-2005," *ENVIRONMENTAL MONITORING AND ASSESSMENT*, vol. 150, no. 1-4, pp. 237–249, 2009.

[53] J. Yin, Z. Yin, H. Zhong, S. Xu, X. Hu, J. Wang, and J. Wu, "Monitoring urban expansion and land use/land cover changes of shanghai metropolitan area during the transitional economy (1979-2009) in china," *ENVIRONMENTAL MONITORING AND ASSESSMENT*, vol. 177, no. 1-4, pp. 609–621, 2011.

[54] G. Lan, L. Ma, Y. Li, and B. Liu, "Mechanism and look-alikes analysis of oil spill monitoring with optical remote sensing," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 8006, 2011, p. 800628 (7 pp.).

[55] D. F. d. A. Lopes, G. L. B. Ramalho, F. N. S. de Medeiros, R. C. S. Costa, and R. T. S. Araujo, "Combining features to improve oil spill classification in sar images," in *STRUCTURAL, SYNTACTIC, AND STATISTICAL PATTERN RECOGNITION, PROCEEDINGS*, vol. 4109, 2006, pp. 928–936.

[56] X. Zhang, M. Pamer, and N. Duke, "Lithologic and mineral information extraction for gold exploration using aster data in the south chocolate mountains (california)," *ISPRS journal OF PHOTOGRAMMETRY AND REMOTE SENSING*, vol. 62, no. 4, pp. 271–282, 2007.

[57] E. Ben-Dor, S. Chabrillat, J. A. M. Dematte, G. R. Taylor, J. Hill, M. L. Whiting, and S. Sommer, "Using imaging spectroscopy to study soil properties," *REMOTE SENSING OF ENVIRONMENT*, vol. 113, pp. S38–S55, 2009.

[58] M. Chica-Olmo and F. Abarca-Hernandez, "Computing geostatistical image texture for remotely sensed data classification," *COMPUTERS & GEOSCIENCES*, vol. 26, no. 4, pp. 373–383, 2000.

[59] G. Metternicht and J. Zinck, "Remote sensing of soil salinity: potentials and constraints," *REMOTE SENSING OF ENVIRONMENT*, vol. 85, no. 1, pp. 1–20, 2003.

[60] F. Zhang, J. Li, Q. Shen, B. Zhang, C. Wu, Y. Wu, G. Wang, S. Wang, and Z. Lu, "Algorithms and schemes for chlorophyll a estimation by remote sensing and optical classification for turbid lake taihu, china," *IEEE journal OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING*, vol. 8, no. 1, pp. 350–364, 2015.

[61] B. Raup, A. Kaeaeb, J. S. Kargel, M. P. Bishop, G. Hamilton, E. Lee, F. Paul, F. Rau, D. Soltesz, S. J. S. Khalsa, M. Beedle, and C. Helm, "Remote sensing and gis technology in the global land ice measurements from space (glims) project," *COMPUTERS & GEOSCIENCES*, vol. 33, no. 1, pp. 104–125, 2007.

[62] P. S. Thenkabail, C. M. Biradar, P. Noojipady, V. Dheeravath, Y. Li, M. Velpuri, M. Gumma, O. R. P. Gangalakunta, H. Turral, X. Cai, J. Vithanage, M. A. Schull, and R. Dutta, "Global irrigated area map (giam), derived from remote sensing, for the end of the last millennium," *INTERNATIONAL journal OF REMOTE SENSING*, vol. 30, no. 14, pp. 3679–3733, 2009.

[63] C. Wright and A. Gallant, "Improved wetland remote sensing in yellowstone national park using classification trees to combine tm imagery and ancillary environmental data," *REMOTE SENSING OF ENVIRONMENT*, vol. 107, no. 4, pp. 582–605, 2007.

[64] F. Frappart, F. Seyler, J. Martinez, J. Leon, and A. Cazenave, "Floodplain water storage in the negro river basin estimated from microwave remote sensing of inundation area and water levels," *REMOTE SENSING OF ENVIRONMENT*, vol. 99, no. 4, pp. 387–399, 2005.

[65] B. Dixon, "Applicability of neuro-fuzzy techniques in predicting ground-water vulnerability: a gis-based sensitivity analysis," *journal OF HYDROLOGY*, vol. 309, no. 1-4, pp. 17–38, 2005.

[66] D. Mandl, S. Frye, P. Cappelaere, M. Handy, F. Policelli, M. Katjizeu, G. Van Langenhove, G. Aube, J. Saulnier, R. Sohlberg, J. Silva, N. Kussul, S. Skakun, S. Ungar, R. Grossman, and J. Szarzynski, "Use of the earth observing one (eo-1) satellite for the namibia sensorweb flood early warning pilot," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE journal of*, vol. 6, no. 2, pp. 298–308, 2013.

[67] J. Blower, A. Gemmell, G. Griffiths, K. Haines, A. Santokhee, and X. Yang, "A web map service implementation for the visualization of multidimensional gridded environmental data," *Environmental Modelling & Software*, vol. 47, no. 0, pp. 218 – 224, 2013.

[68] O. Zavala-Romero, A. Ahmed, E. Chassignet, J. P. Zavala-Hidalgo, A. F. Eguiarte, and A. Meyer-Baese, "An open source java web application to build self-contained web gis sites," *Environmental Modelling and Software*, vol. 62, no. Complete, pp. 210–220, 2014.

[69] N. R. Swain, K. Latu, S. D. Christensen, N. L. Jones, E. J. Nelson, D. P. Ames, and G. P. Williams, "A review of open source software solutions for developing water resources web applications," *Environmental Modelling & Software*, vol. 67, no. 0, pp. 108 – 117, 2015.

[70] S. Martinis, A. Twele, C. Strobl, J. Kersten, and E. Stein, "A multi-scale flood monitoring system based on fully automatic modis and terrasar-x processing chains," *Remote Sensing*, vol. 5, no. 11, p. 5598, 2013.

[71] A. Ferran, S. Bernabe, P. G. Rodriguez, and A. Plaza, "A web-based system for classification of remote sensing data," *IEEE journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 4, pp. 1934–1948, 2013.

[72] S.N.V.Kalluri, Z.Zhang, J.Jájá, S.Liang, and J.R.G.Townshend, "Characterizing land surface anisotropy from avhrr data at a global scale using high performance computing," *International journal of Remote Sensing*, vol. 22, no. 11, pp. 2171–2191, 2001.

[73] A. J. Plaza and C.-I. Chang, *High Performance Computing in Remote Sensing*. Chapman & Hall/CRC, 2007.

[74] D. Kao, R. Bergeron, and T. Sparr, "Efficient proximity search in multivariate data," in *Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on*, Jul 1998, pp. 145–154.

[75] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[76] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[77] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, 2010, pp. 1–10.

[78] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2012, pp. 2–2.

[79] M. Datcu, H. Daschiel, A. Pelizzari, M. Quartulli, A. Galoppo, A. Colapicchioni, M. Pastori, K. Seidel, P. G. Marchetti, and S. dÉlia, "Information mining in remote sensing image archives: system concepts." *IEEE T. Geoscience and Remote Sensing*, vol. 41, no. 12, pp. 2923–2936, 2003.

[80] N. Chauffert, J. Israel, and B. Le Saux, "Boosting for interactive man-made structure classification," in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*. IEEE, 2012, pp. 6856–6859.

[81] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 12, pp. 1349–1380, 2000.

[82] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys CSUR*, vol. 40, no. 2, p. 5, 2008.

[83] J. M. Peña-Barragán, M. K. Ngugi, R. E. Plant, and J. Six, "Object-based crop identification using multiple vegetation indices, textural features and crop phenology," *Remote Sensing of Environment*, vol. 115, no. 6, pp. 1301 – 1316, 2011.

[84] M. Blume and D. R. Ballard, "Image annotation based on learning vector quantization and localized haar wavelet transform features," in *Proc. SPIE 3077 181–190*, 1997, pp. 181–190.

[85] I. G. Olaizola, M. Quartulli, J. Florez, and B. Sierra, "Trace transform based method for color image domain identification," *Multimedia, IEEE Transactions on*, vol. 16, no. 3, pp. 679–685, 2014.

[86] I. G. Olaizola, G. Marcos, P. Kramer, J. Florez, and B. Sierra, "Architecture for semi-automatic multimedia analysis by hypothesis reinforcement," in *Broadband Multimedia Systems and Broadcasting, 2009. BMSB'09. IEEE International Symposium on*. IEEE, 2009, pp. 1–6.

[87] A. Talib, M. Mahmuddin, H. Husni, and L. E. George, "A weighted dominant color descriptor for content-based image retrieval," *journal of Visual Communication and Image Representation*, vol. 24, no. 3, pp. 345 – 360, 2013.

[88] M. Tkalcic and J. Tasic, "Colour spaces: perceptual, historical and applicational background," in *EUROCON 2003. Computer as a Tool. The IEEE Region 8*, vol. 1, Sept 2003, pp. 304–308 vol.1.

[89] O. D. Faugeras and W. K. Pratt, "Decorrelation methods of texture feature extraction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-2, no. 4, pp. 323–332, 1980.

[90] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893.

[91] E. P. Fotiadis, M. Garzón, and A. Barrientos, "Human detection from a mobile robot using fusion of laser and vision information," *Sensors*, vol. 13, no. 9, 2013.

[92] B. A. Rosdi, C. W. Shing, and S. A. Suandi, "Finger vein recognition using local line binary pattern," *Sensors*, vol. 11, no. 12, pp. 11 357–11 371, 2011.

[93] C. A. Hlavka, "Land-use mapping using edge density texture measures on thematic mapper simulator data," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. GE-25, no. 1, pp. 104–108, 1987.

[94] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: a line segment detector," *Image Processing Online*, vol. 2, pp. 35–55, 2012.

[95] E. Ardizzone, A. Bruno, and G. Mazzola, "Visual saliency by keypoints distribution analysis," in *Image Analysis and Processing–ICIAP 2011*. Springer, 2011, pp. 691–699.

[96] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[97] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[98] W. Jia, X. He, H. Zhang, and Q. Wu, "Combining edge and colour information for number plate detection," in *Proceedings of Image and Vision Computing New Zealand 2007*. VLDB Endowment, 2007, pp. 227–232.

[99] X. Perrotton, M. Sturzel, and M. Roux, "Automatic object detection on aerial images using local descriptors and image synthesis," in *Computer Vision Systems*. Springer, 2008, pp. 302–311.

[100] J. Inglada and E. Christophe, "The orfeo toolbox remote sensing image processing software," in *Geoscience and Remote Sensing Symposium,2009 IEEE International, IGARSS 2009*, vol. 4, 2009, pp. IV–733–IV–736.

[101] E. Pasolli, F. Melgani, N. Alajlan, and N. Conci, "Optical image classification: A ground-truth design framework," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 51, no. 6, pp. 3580–3597, 2013.

[102] M. Khalefa, M. F. Mokbel, and J. Levandoski, "Skyline query processing for uncertain data," in *In Proceedings of the ACM International Conference on Information and Knowledge Management, ACM CIKM 2010*. ACM, 2010.

[103] B. Yang, H. Lu, and C. S. Jensen, "Probabilistic threshold k nearest neighbor queries over moving objects in symbolic indoor space," in *Proceedings of the 13th International Conference on Extending Database Technology*. New York, NY, USA: ACM, 2010, pp. 335–346.

[104] S. Maneewongvatana and D. M. Mount, "On the efficiency of nearest neighbor searching with data clustered in lower dimensions." in *International Conference on Computational Science (1)*, vol. 2073. Springer, 2001, pp. 842–851.

[105] J. Lozano, M. Quartulli, I. Tamayo, M. Laka, and I. G. Olaizola, "Visual analytics for built-up area understanding from metric resolution earth observation data," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2013.

[106] J. Lozano, N. Aginako, M. Quartulli, and I. G. Olaizola, "Semi automatic remote sensing image layer generator based on web based visual analytics," *5th Jubilee International Conference on Cartography and GIS*, 2014.

[107] M. Molinier, J. Laaksonen, and T. Hame, "Detecting man-made structures and changes in satellite imagery with a content-based information retrieval

system built on self-organizing maps," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, no. 4, pp. 861–874, 2007.

[108] J. Lozano Silva, N. Aginako Bengoa, M. Quartulli, I. Olaizola, and E. Zulueta, "Web-based supervised thematic mapping," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. PP, no. 99, pp. 1–12, 2015.

[109] J. Lozano, N. Aginako, M. Quartulli, I. G.Olaizola, and E. Zulueta, "Large scale thematic mapping by supervised machine learning on 'big data'distributed cluster computing frameworks," in *Geoscience and Remote Sensing Symposium, 2015. IGARSS '15. Proceedings. 2015 IEEE International*, 2015, p. 4 pp.

[110] J. Lozano, N. Aginako, M. Quartulli, I. G. Olaizola, and E. Zulueta, "Scalable machine learning for fast thematic mapping in web server," in *Proceedings of the 2014 conference on Big Data from Space (BiDS '14)*. Publications Office of the European Union, 2014, pp. 38–41.

[111] M. Quartulli, J. Lozano, N. Aginako, and I. G.Olaizola, "Beyond the lambda architecture: Effective scheduling for large scale eo information mining and interactive thematic mapping," in *Geoscience and Remote Sensing Symposium, 2015. IGARSS '15. Proceedings. 2015 IEEE International*, 2015, p. 4 pp.

[112] J. Arocena, J. Lozano, N. Aginako, M. Quartulli, I. G.Olaizola, and J. Bermudez, "Linked open data for raster and vector geospatial information processing," in *Geoscience and Remote Sensing Symposium, 2015. IGARSS '15. Proceedings. 2015 IEEE International*, 2015, p. 4 pp.

[113] M. Quartulli, M. Zorrilla, and I. García, "On the image content of the esa eusc jrc workshop on image infomation mining," in *2012 ESA-E*, 2012.

[114] P. C. Wong and J. Thomas, "Visual analytics," *Computer Graphics and Applications, IEEE*, vol. 24, no. 5, pp. 20–21, 2004.

[115] E. Plugge, P. Membrey, and T. Hawkins, "Introduction to mongodb," *The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing*, pp. 3–17, 2010.

[116] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, 2000.

[117] S. Chaudhuri and U. Dayal, "An Overview of Data Warehousing and OLAP Technology," *SIGMOD Rec.*, vol. 26, no. 1, pp. 65–74, Mar. 1997.