



AVANCES EN ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES

Editado por
Denisa-Andreea Constantinescu
Sonia González Navarro
Gerardo Bandera Burgueño
José María González Linares
Julián Ramos Cózar
Andrés Rodríguez Moreno
Jose Manuel Palomares Muñoz
Antonio Martínez Álvarez



Avances en arquitectura y tecnología de computadores

Actas de las Jornadas SARTECO 20/21

Málaga, 21 a 24 de Septiembre de 2021

Avances en arquitectura y tecnología de computadores
Actas de las Jornadas SARTECO 20/21

Editores: Denisa-Andreea Constantinescu, Sonia González Navarro, Gerardo Bandera Burgueño, José María González Linares, Julián Ramos Cózar, Andrés Rodríguez Moreno, Jose Manuel Palomares Muñoz, Antonio Martínez Álvarez

(c) 2021, Jornadas SARTECO

ISBN-13: 978-84-09-32487-3

Málaga, 2021

ISBN 978-840932487-3



9 788409 324873

Comités de coordinación

Comité de Dirección de las Jornadas SARTECO

Francisco Quiles Flor (UCLM, Presidente)
Víctor Viñals Yúfera (UNIZAR, Vicepresidente)
Gracia Ester Martín Garzón (UAL, Secretaria)
Francisco Tirado Fernández (UCM, Presidente de Honor)

Comité de Organización JP20/21

Mª Ángeles González Navarro (UMA)
Sonia González Navarro (UMA)
Denisa-Andreea Constantinescu (UMA)
Andrés Rodríguez Moreno (UMA)
Julián Ramos Cózar (UMA)
José María González Linares (UMA)
Francisco Corbera Peña (UMA)
Gerardo Bandera Burgueño (UMA)
Rafael Asenjo Plaza (UMA)

Comité de Organización JCER20/21

Antonio Martínez-Álvarez (UA)
Jesús González Peñalver (UGR)
Sergio Cuenca-Asensi (UA)
Miguel A. Vega Rodríguez (UEX)
Miguel Damas Hermoso (UGR)
Jose Manuel Palomares Muñoz (UCO)
Joaquín Olivares Bueno (UCO)
Jesús Barba Romero (UCLM)
Gustavo Sutter (UAM)
Ignacio Bravo-Muñoz (UAH)
José Torres País (UV)
Jordi Carrabina Bordoll (UAB)
Juan Suardíaz Muro (UPCT)
Goiuria Sagardui Mendieta (UMondragon)
Gabriel Noé Mujica Rojas (UPM)

Comité T3M20/21 y TFM3M20/21

Francisco Quiles Flor (UCLM) (Presidente)
Gracia Ester Martín Garzón (UAL) (Secretaria)
Francisco Tirado Fernández (Presidente de Honor)
Enrique S. Quintana Ortí (UJI) (Vocal, Junta directiva SARTECO)
Miquel Moretó Planas (UPC) (Vocal, Junta directiva SARTECO)

Indice

Parte 1.- XXXI Jornadas de Paralelismo

Aplicaciones de la computación de altas prestaciones

De Kepler a Turing: Evaluando Arquitecturas GPU para Acelerar el Algoritmo de Fitch <i>Sergio Santander-Jiménez, Miguel A. Vega-Rodríguez, Antonio Zahinos-Márquez, Leonel Souza</i>	5
Using the wavelet transform for DMR detection and visualization <i>Lisardo Fernández, Mariano Pérez, Juan M. Orduña, Ricardo Olanda</i>	13
Influencia del algoritmo de segmentación en la precisión y el coste computacional de la detección de cambios multiclasa sobre imágenes hiperespectrales <i>F. Javier Cardama, Álvaro G. Dieste, Alberto S. Garea, Dora Blanco Heras, Francisco Argüello</i>	23
Paralelización de un algoritmo de optimización multi-objetivo aplicado al problema de resumen extractivo multi-documento <i>Jesus M. Sanchez-Gomez, Miguel A. Vega-Rodríguez, Carlos J. Pérez</i>	31
Aceleración de planes de radioterapia de intensidad modulada basados el método de Descenso por Gradiente <i>Juan José Moreno Riado, Ester Martín Garzón, Leocadio González Casado, Janusz Miroforidis, Ignacy Kaliszewski</i>	41
Una nueva técnica de aumento de datos para la clasificación de imágenes hiperespectrales mediante CNN <i>Álvaro Acción Montes, Dora Blanco Heras, Francisco Argüello</i>	49

Arquitecturas heterogéneas y modelos de programación para GPU, FPGA y aceleradores de IA

Estudio y Análisis de la Simulación de Sistemas Eléctricos de Potencia con Generación de Energía Renovable Mediante Tarjetas Gráficas <i>Alberto Jiménez-Ruiz, Miguel Cañas-Carretón, Gerardo Fernández-Escribano, José L. Sánchez, Hari Kalva</i>	59
---	----

Towards high-level heterogeneous co-execution via oneAPI <i>Raúl Nozal, Jose Luis Bosque</i>	69
---	----

SkyFlow: Streaming heterogeneo para calculo de skyline mediante FlowGraph y oneAPI <i>Jose Carlos Romero, Felipe Muñoz López, Antonio Vilches Reina, Andres Rodríguez Moreno, Angeles Navarro, Rafael Asenjo</i>	79
---	----

PyDTNN: Entorno para Entrenamiento e Inferencia con Redes Neuronales Profundas <i>Sergio Barrachina, Adrián Castelló, Mar Catalán, Manuel F. Dolz, Jose I. Mestre, Cristian Ramírez, David Rodríguez</i>	89
---	----

Estado del arte de los lenguajes de descripción de hardware: un enfoque práctico <i>Francisco Jiménez-Fiérrez, Alberto Jiménez-Ruiz, Damián Ruiz-Coll, Gerardo Fernández-Escribano, Pedro Cuenca, Hari Kalva</i>	97
---	----

Influencia del algoritmo de segmentación en la precisión y el coste computacional de la detección de cambios multiclase sobre imágenes hiperespectrales

F. Javier Cardama¹, Álvaro G. Dieste¹, Alberto S. Garea¹, Dora B. Heras¹, y Francisco Argüello²

Resumen— En el campo de la teledetección es habitual la necesidad de detectar cambios a partir de imágenes hiperespectrales de una misma área geográfica capturadas en diferentes momentos. Una gran parte de las técnicas de detección de cambios permiten además de detectar los cambios, clasificar los tipos de cambio que se han producido. El uso de técnicas de extracción de información espacial y espectral sobre áreas de la imagen, más allá del ámbito del píxel, como las técnicas de segmentación, permiten mejorar la calidad de la detección a cambio de un incremento en coste computacional. La elección adecuada de algoritmos para las diferentes etapas implicadas en el proceso y su adecuada programación permiten encontrar una solución de compromiso entre coste computacional y calidad de la detección. En este artículo proponemos y analizamos diferentes alternativas para las etapas de segmentación de las imágenes y clasificación para un método de detección de cambios, haciendo hincapié en el coste computacional de las alternativas.

Palabras clave— detección de cambios, hiperespectral, teledetección, segmentación, clasificación, GPU, OpenMP

I. INTRODUCCIÓN

La creciente disponibilidad de imágenes obtenidas mediante sofisticados sensores situados en satélites ofrecen una fuente de datos que permiten una observación de la superficie de la tierra más detallada que años atrás. Los avances en este tipo de sensores han permitido también ampliar el rango del espectro electromagnético capturado facilitando así las diferentes tareas en las que se emplean imágenes de teledetección, en particular, multi e hiperespectrales, tales como clasificación de escenas, agricultura de precisión, registrado de imágenes o detección de cambios, entre otras [1], [2].

La detección de cambios (*CD*, *Change Detection*) basada en datos de teledetección [3] es un método ampliamente usado en monitorización del entorno, evaluación de desastres, revisión de mapas, análisis de la deforestación, etc. El objetivo fundamental es el de identificar las diferencias producidas en un área

¹Centro Singular de Investigación en Tecnologías Inteligentes (CiTIUS), Universidad de Santiago de Compostela, e-mail: {javier.cardama, alvaro.goldar.dieste, jorge.suarez.garea, dora.blanco}@usc.es.

²Dpto. de Electrónica e Computación, Universidad de Santiago de Compostela, España, e-mail: francisco.arguello@usc.es.

mediante la observación de la misma en diferentes períodos de tiempo [4] y requiere la existencia de imágenes multitemporales del área que se pretende analizar. Hoy en día esto no es un problema ya que se ha incrementado notablemente el número de satélites que permiten capturar largas series de imágenes temporales. Además, la proliferación de los vehículos aéreos no tripulados (*UAVs*, *Unmanned Aerial Vehicles*) permite aplicar las técnicas de *CD* sobre zonas de interés de las cuales no se disponga de imágenes de satélite.

Dentro del campo de la detección de cambios, las diferentes técnicas de análisis de imágenes multitemporales se pueden dividir en dos grandes grupos [5]: las que se basan en la fusión de información multitemporal a nivel de característica y las que lo hacen a nivel de decisión. Para la primera aproximación, que es la analizada en este artículo, se realiza una fusión de las imágenes antes de proceder a la detección binaria que indica si un píxel ha sufrido cambio o no. La fusión de información se realiza normalmente mediante operaciones algebraicas como la diferencia entre imágenes, la diferencia entre índices de vegetación [6] o cualquier otra basada en el análisis de vectores de cambio (*CVA*, *Change Vector Analysis*) [7].

De las imágenes multidimensionales, además de la propia información espectral que se puede extraer a nivel de cada píxel individual, también podemos extraer información espacial que puede ser utilizada para detectar estructuras homogéneas según el criterio que se elija. Los algoritmos de segmentación realizan esta tarea y permiten reducir el coste computacional de algunas operaciones que se realizan sobre la imagen [8]. La segmentación forma parte del esquema de detección de cambios presentado en este trabajo.

Los algoritmos de segmentación basados en similaridad agrupan los diferentes píxeles que componen la imagen según sus propiedades, asignando aquellos con características semejantes al mismo segmento. La transformada watershed [9] es un algoritmo no supervisado ampliamente utilizado para segmentación de imágenes multi e hiperespectrales. Dado que este algoritmo no es aplicable directamente sobre imágenes multidimensionales, se debe aplicar un algoritmo previo para reducir la dimensionalidad de la imagen a una única banda manteniendo la mayor cantidad

de información posible. RCMG [10] es un ejemplo de este tipo de algoritmos. Otros de los algoritmos de segmentación ampliamente utilizados en el procesado de imágenes multi e hiperespectrales son los basados en superpíxeles, como SLIC [11], waterpixel [12] o ETPS [13]. Estos generan un mapa de segmentos de la imagen con regularidad y tamaño máximo de los segmentos que puede ser decidido por el usuario. Son algoritmos que suelen ofrecer buenos resultados sobre imágenes hiperespectrales.

Para extraer información espacial a partir de una imagen también son comunes los perfiles morfológicos (*MPs, Morphological Profiles*) [14]. Mediante el uso de un elemento de análisis denominado elemento estructurante (*SE, Structuring Element*), los perfiles eliminan objetos que no encajan dentro de dicho SE. Aplicando de forma sucesiva operaciones morfológicas en las cuales se varía el tamaño del SE, obtendremos una representación de la imagen con varios niveles de detalle. En el caso de imágenes multidimensionales, aplicando MP a cada una de las bandas de una imagen hiperespectral obtenemos un perfil morfológico extendido (*EMP, Extended Morphological Profile*) [15]. Esta etapa de extracción de información espacial, que tiene un alto coste computacional, la utilizamos también en la técnica de detección de cambios propuesta.

Las técnicas que permiten detectar tipos de cambios suelen terminar en una etapa de clasificación. Durante este proceso, a cada uno de los píxeles de la imagen se le asigna una etiqueta de entre un conjunto de clases previamente definido. La clasificación se puede llevar a cabo de forma supervisada o no supervisada. En el primer caso, el algoritmo necesita de cierta información de referencia, es decir, que indique a qué clase pertenecen algunos de los píxeles de la imagen. Esta información será usada durante un proceso de aprendizaje el cual permitirá adquirir el conocimiento necesario para ser capaz de clasificar el resto de la imagen. Esta es la aproximación que seguiremos.

En particular, en este artículo hemos seleccionado dos técnicas de clasificación supervisada. La primera está basada en máquinas de soporte vectorial (*SVMs, Support Vector Machines*) [16] y la segunda en aprendizaje automático. En nuestro caso, como representante de esta última hemos seleccionado una red neuronal convolucional (*CNN, Convolutional Neural Network*) [17].

El algoritmo SVM fue desarrollado originalmente para la clasificación binaria, pero con el tiempo su uso se ha extendido a problemas de clasificación multiclase y de regresión. En su forma binaria, el algoritmo trata de encontrar el hiperplano óptimo que sea capaz de separar dos muestras de entrenamiento pertenecientes a clases distintas, maximizando la distancia entre los puntos más cercanos de cada clase.

Por su parte, una CNN es una red neural formada básicamente por un conjunto de filtros convolucionales anidados que extraen características de forma jerárquica permitiendo aprender o representar dichas

características a diferentes niveles de complejidad, donde los conceptos más generales son definidos en relación a conceptos más simples y las representaciones más abstractas son obtenidas en base a otras que lo son menos [18], [17]. El método estándar de aprendizaje para redes CNN consiste en un proceso iterativo, el cual se repite hasta alcanzar un determinado número de iteraciones o hasta conseguir reducir por debajo de un determinado umbral el error producido por la red, es decir, la diferencia entre la salida obtenida por la red y la salida deseada.

La gran cantidad de información que se maneja cuando se trabaja con imágenes hiperespectrales implica un alto coste computacional. El uso de diferentes técnicas de paralelización como pueden ser la utilización de la API de programación paralela OpenMP [19] o la ejecución de los algoritmos sobre dispositivos específicos como las unidades de procesamiento gráfico (*GPUs, Graphics Processing Units*) [20], [21], permiten reducir ese coste computacional.

En este artículo se analizan diferentes configuraciones para una técnica de CD multiclase sobre imágenes multidimensionales, variando los algoritmos de segmentación y clasificación utilizados, y evaluando el coste computacional de las propuestas. Se presentan además dos implementaciones, una secuencial y otra en la que algunas secciones han sido programadas en OpenMP y GPU para extraer el máximo rendimiento de un ordenador de sobremesa.

El resto del artículo se organiza como sigue: La Sección II describe las diferentes partes del esquema, así como los algoritmos utilizados. En la Sección III se detallan las características tanto del hardware utilizado como de los conjuntos de datos empleados en los experimentos. La evaluación se realiza en la Sección IV. Y, finalmente, la Sección V presenta las conclusiones.

II. TÉCNICA DE DETECCIÓN DE CAMBIOS

En esta sección se describen las diferentes etapas de la técnica de detección de cambios multiclase sobre imágenes multidimensionales analizada en este artículo (ver Fig. 1), que fue presentada en [22] y sobre la que se realizarán diferentes mejoras. El objetivo es detectar cambios entre dos imágenes hiperespectrales obtenidas en diferentes instantes de tiempo indicando a qué clase de cambio pertenece cada píxel. Las etapas son las siguientes:

A. Mapa binario de cambios espectral-espacial

A partir de las dos imágenes de partida se produce un mapa binario de cambios que permite filtrar todos aquellos píxeles de la imagen original en los que se haya producido algún tipo de cambio y que, por lo tanto, serán los que interese clasificar para decidir a qué clase de cambio pertenecen.

El proceso se realiza a nivel de zonas uniformes o segmentos. Para extraer este tipo de información espacial podemos emplear diferentes algoritmos de segmentación. En este artículo proponemos diferentes

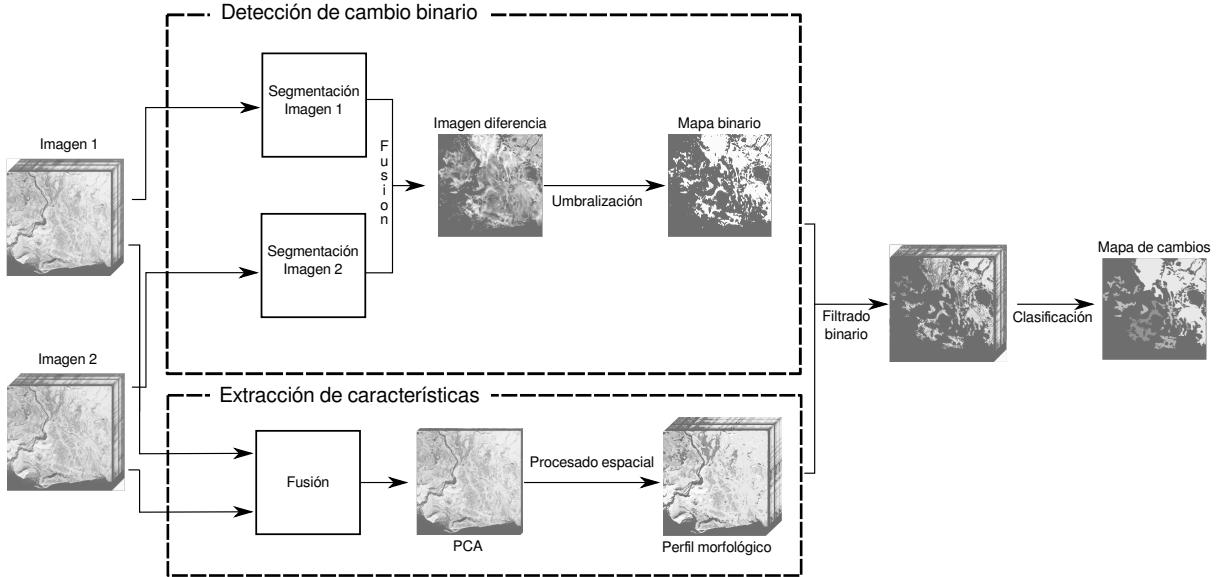


Fig. 1: Técnica de detección de cambios multiclase sobre imágenes hiperespectrales.

algoritmos de segmentación alternativos: watershed [9], waterpixel [12] y SLIC [11]. Los dos últimos son algoritmos de segmentación basados en superpíxeles, es decir, en estructuras cuya regularidad y tamaño pueden ser controladas durante la evolución del algoritmo.

Con las imágenes segmentadas, el siguiente paso es su fusión, para lo cual se emplea una técnica basada en CVA que calcula la distancia y el ángulo entre pares de píxeles correspondientes a la misma posición en ambas imágenes. En concreto, se usa la distancia SAM (*Spectral Angle Mapper*):

$$\alpha_{i,j} = \frac{2}{\pi} \cos^{-1} \left(\frac{S_j \cdot S_i}{\|S_j\| \|S_i\|} \right) \in [0, 1], \quad (1)$$

donde i y j son los dos píxeles bajo consideración y $\alpha_{i,j}$ es el ángulo espectral entre el espectro del píxel i (S_i) con respecto al espectro del píxel j (S_j).

A continuación, como paso final para la obtención del mapa binario de detección de cambios, se aplica una umbralización sobre la imagen diferencia mediante el algoritmo de Otsu que, en su forma más sencilla, devuelve un único umbral de intensidad que separa los píxeles en dos clases, primer plano y fondo [23]. Este proceso requiere de la generación del histograma de la imagen fusionada que contiene tantos rangos de valores como el máximo valor de diferencia en la imagen, para mantener la mayor cantidad de información posible. El valor de umbralización óptimo se calcula iterativamente y es el que proporciona la mínima varianza intraclasa, definida como la suma de las varianzas de las dos clases:

$$\sigma_w^2(t) = w_0(t)\sigma_0^2(t) + w_1(t)\sigma_1^2(t), \quad (2)$$

siendo σ_0^2 y σ_1^2 las varianzas de las dos clases cuyas probabilidades, separadas por el umbral t , son w_0 y w_1 . A continuación, se procesa la imagen fusionada de forma que aquellos píxeles con valor superior al umbral son etiquetados como píxeles de cambio. El proceso puede producir píxeles no conectados o ruido

en el mapa de cambios binario, que se corrige aplicando un procesamiento de regularización espacial iterativo que explota la información de los vecinos más cercanos.

B. Extracción de características

El objetivo de esta rama de la técnica es el de extraer la información espacial y espectral más relevante de las imágenes originales. Primero se fusionan ambas imágenes al igual que se hizo para la rama de detección de cambios binarios. A continuación se realiza una reducción de dimensionalidad con el algoritmo PCA [24], lo que permite extraer las características más relevantes de la imagen fusionada.

Finalmente, se calcula el perfil morfológico extendido sobre el resultado de PCA. El MP fue introducido en [25] y posteriormente extendido a imágenes multidimensionales en [26] para extraer información sobre el contraste y el tamaño de las estructuras presentes en la imagen. El MP de orden n de la imagen I se puede expresar como:

$$\text{MP}^{(n)}(I) = \{\gamma_r^{(n)}(I), \dots, \gamma_r^{(1)}(I), I, \phi_r^{(1)}(I), \dots, \phi_r^{(n)}(I)\}, \quad (3)$$

siendo $\gamma_r^{(i)}$ y $\phi_r^{(i)}$ los operadores de apertura y cierre por reconstrucción respectivamente, con un elemento estructural i desde 1 a n , cuyo tamaño se incrementa normalmente en pasos de 1 o 2. Al variar el tamaño del elemento estructural se cambia la escala a la que se consideran las estructuras espaciales.

Cuando se usa la aproximación de MP sobre los datos hiperespectrales, se usan los m PCs más significativos como imagen base. El resultado es un EMP,

$$\text{EMP}_m^{(n)}(I) = \{(MP)_1^{(n)}(I), \dots, (MP)_m^{(n)}(I)\}, \quad (4)$$

con $m(2n + 1)$ componentes.

C. Clasificación

Tras la extracción de características se filtra la información usando el mapa de cambios binario, de modo que tan solo se pasa a la clasificación el conjunto de píxeles identificados como cambios. El filtrado contribuye también a la reducción de coste computacional del proceso de clasificación, ya que se reduce el número de muestras que deben ser clasificadas.

En este artículo se emplean dos clasificadores supervisados distintos: uno basado en técnicas estadísticas, en particular, en máquinas de soporte vectorial (SVM), y otro basado en técnicas de aprendizaje automático, una red neuronal convolucional (CNN).

III. MÉTODOS Y HERRAMIENTAS

En esta sección se detallarán tanto las características del hardware utilizado en los experimentos como las medidas de precisión y tiempo utilizadas para evaluarlos. También se describirán los conjuntos de datos utilizados en los experimentos.

La técnica propuesta ha sido evaluada en un PC con un hexa-core Intel Core i7-8750H a 2.20 GHz, 16 GB de memoria RAM y Ubuntu 20.04.1 LTS. El código se ha compilado usando gcc 9.3.0 con soporte para OpenMP (OMP) 4 bajo Linux para algunos de los códigos, y las librerías LIBSVM 3.25 y LAPACKE 3.9.0.

En cuanto a la implementación en GPU, se ha utilizado una tarjeta NVIDIA GeForce RTX 2060 con 30 SM y un total de 1920 procesadores CUDA. Se usó la versión 11.2.67 del conjunto de herramientas CUDA para Linux con soporte para cuDNN, y las librerías PyTorch 1.8.0, NumPy 1.19.2 y scikit-learn 0.24.1.

La precisión de los resultados obtenidos en los distintos experimentos está expresada en términos de precisión total (OA), que es el porcentaje de píxeles correctamente clasificados, precisión media (AA), que se obtiene como la media de las precisiones de las diferentes clases, y el coeficiente kappa [27], que corresponde al porcentaje de acierto corregido por la cantidad de acierto que sería de esperar debido al azar. Todos los resultados han sido calculados como el valor medio de 10 ejecuciones independientes.

A. Conjunto de datos

La técnica ha sido evaluada sobre la escena de Hermiston que consiste en dos imágenes obtenidas por el sensor HYPERION sobre un área de la ciudad de Hermiston, Oregón ($N 45^{\circ}58'49.4''$, $W 119^{\circ}13'09.6''$), en diferentes fechas. El conjunto de datos muestra un área rural con varias zonas de cultivo circulares. Las imágenes corresponden a los años 2004 y 2007 y fueron descargadas de la web del servicio geológico de Estados Unidos (*USGS, United States Geological Survey*). Las dimensiones espaciales de las imágenes son 390×200 píxeles con una resolución espacial de

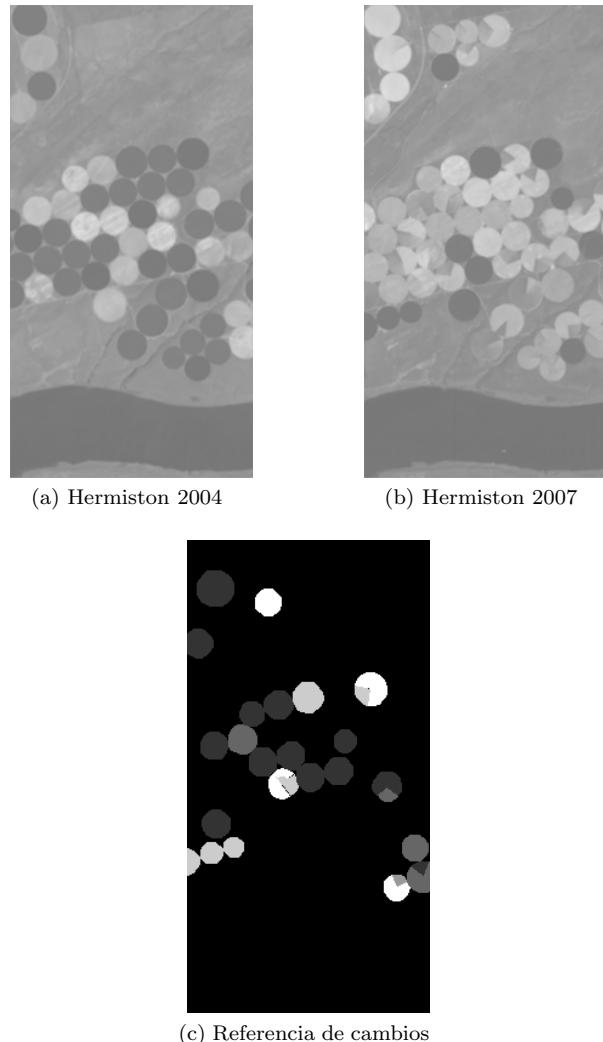


Fig. 2: Escena de Hermiston y mapa de referencia de cambios multiclas.

30 metros por píxel, mientras que su dimensión espectral cubre 242 bandas.

La Figura 2 muestra las dos imágenes y el mapa de referencia disponible donde se han definido cinco clases de cambios de cultivos entre las imágenes de la escena. La Tabla I muestra un resumen del contenido del mapa de referencia de cambios.

Tabla I: Clases de cambios y número de píxeles en cada una de ellas para la escena de Hermiston

Clase	Número de muestras
1	5558
2	1331
3	79
4	1557
5	1461
Total	9986

IV. RESULTADOS EXPERIMENTALES

Esta sección muestra los resultados experimentales obtenidos por las distintas configuraciones de la técnica propuesta, evaluadas en términos de precisión de la detección de cambios a nivel de píxel y de tiempo de ejecución.

En cuanto a los valores de configuración de los

distintos algoritmos, los valores de tamaño y regularidad para el algoritmo SLIC son 5 y 15, respectivamente. Para el algoritmo waterpixel, esos mismos parámetros tienen los valores 5 y 0.45, respectivamente. El parámetro que define los marcadores en el algoritmo watershed está fijado a 0.19 para la imagen de Hermiston 2004, y a 0.22 para la imagen de Hermiston 2007. En los tres algoritmos de segmentación el número de vecinos a tener en cuenta es de 8. Por su parte, los valores para los parámetros C y γ utilizados para el clasificador SVM son 1000 y 1, respectivamente.

En cuanto a la red CNN, se trata de una red con tres capas convolucionales unidimensionales seguidas por una serie de capas completamente conectadas encargadas de la clasificación final. Las dos primeras capas convolucionales van seguidas de una capa de *pooling*, mientras que la tercera va acompañada de una capa de *dropout*. El tamaño de los filtros convolucionales es de 10 elementos, y el número de filtros por capa es de 64, 128 y 256 para la primera, segunda y tercera capa convolucional, respectivamente. La capa de *dropout* está configurada con una probabilidad del 50 %. La sección encargada de la fase final de clasificación está constituida por dos capas completamente conectadas de 256 neuronas la primera y 5 la segunda, coincidiendo esta última con el número de clases a identificar. El valor del ratio de aprendizaje (*learning rate*) es de 0.001. Se ha utilizado el optimizador Adam [28] con los valores β_1 y β_2 configurados como 0.9 y 0.999, respectivamente. Los pesos de la red se han inicializado según el método descrito en [29], mientras que los sesgos se han inicializado a 0. Para la fase de entrenamiento se han empleado 200 épocas con lotes de 32 elementos.

Tabla II: Precisión del mapa binario de cambios obtenido para escena Hermiston (OA en %, MAs y FAs en número de píxeles) para los tres algoritmos de segmentación indicados (columna Seg. en la tabla).

Seg.	OA _{CB}	FAs	MAs
WS	98.26 %	885	473
WP	98.61 %	565	520
SLIC	98.72 %	536	461

La Tabla II muestra las precisiones en porcentaje de acierto, falsas alarmas de cambio (FA) y número de cambios no detectados (MA) para la detección binaria de cambios indicada en la Figura 1, es decir, para producir el mapa que indica si cada píxel ha cambiado o no. Se comparan los resultados utilizando tres algoritmos de segmentación: watershed (WS en la tabla) y los dos algoritmos basados en superpíxeles llamados waterpixel (WP en la tabla) y SLIC. Se observa que la diferencia en precisión entre las tres opciones es muy pequeña. Una de las razones es la homogeneidad de la mayor parte de estructuras presentes en estas imágenes.

Los mapas de segmentación obtenidos para una de las imágenes de la escena se muestran en la Figura 3. Los algoritmos de segmentación han sido adaptados para que segmenten imágenes hiperespectrales apro-

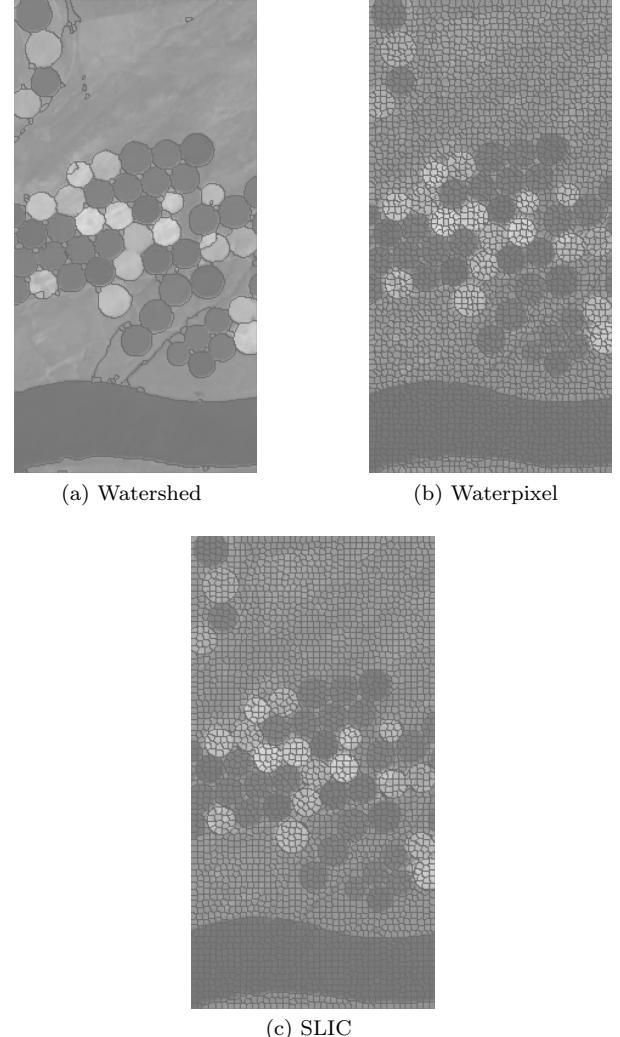


Fig. 3: Mapas de segmentación de Hermiston 2004, una de las imágenes de la escena Hermiston, empleando diferentes algoritmos de segmentación.

vechando toda la información que ellas contienen, es decir, de modo que todas las componentes espectrales de cada pixel son tenidas en cuenta. Como se puede observar, la segmentación se ha realizado a grano fino mediante la búsqueda de los parámetros de configuración de los algoritmos de modo que los bordes de los segmentos coincidiesen lo máximo posible con los bordes visibles en la reconstrucción RGB de las imágenes. La calidad aparente de la segmentación es similar en los tres casos. Sería necesario que las imágenes contasen con estructuras más heterogéneas para poner a prueba a los diferentes algoritmos. El algoritmo SLIC ofrece resultados ligeramente más altos, lo cual es consistente con los buenos resultados para extracción de información espacial que se ha comprobado en la bibliografía que realiza este algoritmo.

Una vez analizada la precisión de la detección binaria de cambios, la Tabla III muestra los valores de precisión de la detección para el proceso completo, es decir, cuando se clasifican los tipos de cambio. En particular, se muestran resultados para las diferentes configuraciones de la técnica considerando los tres algoritmos de segmentación y dos alternativas de

Tabla III: Precisiones de la detección de cambios binaria (CB), clasificación multiclase de los cambios (CM) y global para la técnica en su conjunto para la escena Hermiston. Se comparan diferentes configuraciones de la técnica de detección de cambios. Los resultados están dados en términos de OA, AA y kappa (en %).

Técnica	CB			CM			Global		
	OA	OA	AA	K	OA	AA	K		
WS + SVM	98.26	97.41	95.63	95.89	93.07	92.77	89.26		
WP + SVM	98.61	97.51	95.48	96.05	92.69	92.27	88.70		
SLIC + SVM	98.72	97.48	95.89	96.01	93.24	93.07	89.55		
WS + CNN	98.26	96.38	94.21	94.27	92.19	91.55	87.91		
WP + CNN	98.61	97.15	95.30	95.46	92.38	92.13	88.22		
SLIC + CNN	98.72	96.83	94.91	94.98	92.69	92.21	88.67		

clasificación: SVM y CNN. Se muestran los valores de precisión a nivel de píxel en la detección binaria (CB), la precisión al clasificar los píxeles filtrados como cambios para asignarles una clase de cambio (CM), y finalmente la precisión total de la detección teniendo en cuenta todo el proceso (Global).

Respecto de las diferentes opciones de segmentación, en la tabla se observa que los resultados son muy similares entre sí para esta escena, tanto clasificando con SVM como con la red CNN. Los mejores resultados se obtienen segmentando mediante el algoritmo SLIC la igual que ocurría para la detección binaria, tanto si la clasificación se realiza mediante SVM como si se clasifican los cambios mediante la red CNN. Se observa también que, atendiendo a resultados de precisión, no hay diferencias que justifiquen elegir uno de los métodos de clasificación de entre SVM y CNN.

Tabla IV: Tiempos en segundos para la ejecución secuencial de las diferentes configuraciones de detección de cambios. Escena Hermiston.

Técnica	Seg.	CB	Espacial	CM	Total
WS + SVM	0.66	0.76	27.90	0.47	57.68
WP + SVM	0.54	0.78	27.90	0.46	57.57
SLIC + SVM	2.15	0.78	27.90	0.45	59.17
WS + CNN	0.66	0.76	27.90	159.46	216.16
WP + CNN	0.54	0.78	27.90	159.35	216.59
SLIC + CNN	2.15	0.78	27.90	158.18	216.90

Tabla V: Tiempos en segundos para las diferentes configuraciones de detección de cambios ejecutando EMP en OpenMP (incluida en Espacial) y la clasificación CNN en GPU (incluida en CM para las técnicas que usan CNN). Escena Hermiston.

Técnica	Seg.	CB	Espacial	CM	Total
WS + SVM	0.66	0.76	5.43	0.47	12.75
WP + SVM	0.54	0.78	5.43	0.46	14.24
SLIC + SVM	2.15	0.78	5.43	0.45	12.64
WS + CNN	0.66	0.76	5.43	37.44	49.73
WP + CNN	0.54	0.78	5.43	34.57	48.36
SLIC + CNN	2.15	0.78	5.43	35.20	47.50

Seleccionar la configuración para la detección de cambios de esta técnica requiere considerar no solo la precisión de la detección, sino también el coste computacional.

En la Tabla IV se muestran los tiempos de ejecución para las diferentes etapas del algoritmo cuando es ejecutado de manera secuencial en CPU. Se muestran los tiempos para la segmentación, la detección de cambios binarios (CB), la extracción de informa-

ción mediante EMP (Espacial) y, finalmente, la clasificación de cambios (CM) y el tiempo total de ejecución de la técnica. Se puede observar que las técnicas utilizando la CNN para la clasificación tienen un tiempo de ejecución cuatro veces superior a las que usan SVM. Además, se puede observar que para una escena pequeña como es Hermiston, la etapa más costosa de toda la técnica es el cálculo de la clasificación si esta se realiza mediante la red CNN, seguida por la extracción de información espacial (Espacial en la tabla). Esta última etapa consiste, como se mostraba en la Figura 1), en una etapa de reducción de componentes espectrales para reducir el tamaño del conjunto de datos mediante PCA seguida del cálculo del perfil morfológico (EMP), que es la etapa realmente costosa (consume 99 % del tiempo de la extracción de información espacial).

Para reducir el coste de estas etapas se ha realizado una primera versión mejorada del código. La Tabla V muestra los tiempos de ejecución cuando la red CNN se computa en GPU y el EMP mediante una implementación en OpenMP utilizando 12 hilos. El tiempo total se ha reducido sustancialmente con respecto a la tabla anterior, pero aún así el coste computacional de la clasificación usando la CNN sigue siendo superior, cuatro veces en concreto, en comparación con utilizar SVM.

Para la escena presentada los datos indican que una decisión de compromiso entre tiempo de ejecución y precisión sería usar una configuración en la que se segmenta usando SLIC y se clasifica mediante SVM. Sin embargo, la selección de la mejor configuración de la técnica de detección de cambios requeriría un análisis de un mayor número de escenas correspondientes a diferentes tipos de cambios y con estructuras geométricas más variadas.

Para imágenes de mayor tamaño el tiempo de ejecución de la segmentación subirá sustancialmente. Por ejemplo, para una imagen multiespectral de 5 bandas de superficie terrestre de tamaño 6722×6689 píxeles correspondiente a zonas de vegetación natural al borde de un río y con estructuras espaciales variadas en forma y tamaño, los tiempos que se han obtenido para los 3 algoritmos de segmentación WS, WP y SLIC son de 40.59, 25.19 y 36.14 segundos respectivamente. Se aprecia que para imágenes de mayor tamaño y mayor complejidad en las estructuras espaciales es el algoritmo watershed es el que emplea un mayor tiempo en el proceso de segmentación y, por tanto, es la opción menos recomendable.

V. CONCLUSIONES

En este artículo se analizan diferentes configuraciones de una técnica de detección de cambios multiclase aplicada a pares de imágenes hiperespectrales de cobertura terrestre. La técnica está basada en segmentación y extracción de características. Se parte de la segmentación de ambas imágenes y contiene también una fase de extracción de características donde, después de una primera fase de reducción de dimensionalidad, se extrae información espacial mediante

el uso de EMP. Finalmente, tras una fase de filtrado, se realiza clasificación supervisada generando como salida un mapa binario de cambios.

Se han comparado los tres algoritmos de segmentación watershed, waterpixel y SLIC, y los clasificadores SVM, y una CNN. Los resultados de precisión de la clasificación muestran que, para el conjunto de datos bajo estudio, los diferentes algoritmos de segmentación obtienen precisiones muy similares estando ligeramente por encima el algoritmo SLIC. La clasificación global alcanza el valor máximo de 93.24 % usando SLIC y clasificando mediante SVM.

En cuanto a los tiempos de ejecución, las etapas más costosas de la técnica, en tiempo de ejecución, son las de extracción de información espacial, en particular el cálculo del perfil morfológico, y la clasificación mediante redes CNN. Una primera versión mejorada basada en OpenMP y ejecución en GPU permite una aceleración de aproximadamente 4×. En cuanto al tiempo de segmentación, el más alto corresponde al algoritmo que ofrece mejores resultados: SLIC.

Se plantea como trabajo futuro el análisis detallado de la técnica de detección de cambios sobre escenas de mayor tamaño con estructuras geométricas más variadas. También será necesario abordar el desarrollo de implementaciones paralelas de la técnica que resulten computacionalmente menos costosas.

AGRADECIMIENTOS

Este trabajo ha sido financiado en parte por el Ministerio de Ciencia e Innovación, Gobierno de España (números de subvención PID2019-104834GB-I00), y la Conselleria de Educación, Universidade e Formación Profesional (número de subvención ED431C 2018/19, y acreditación 2019–2022 ED431G-2019/04). Todos ellos están cofinanciados por el Fondo Europeo de Desarrollo Regional (FEDER).

REFERENCIAS

- [1] Mathieu Fauvel, Yuliya Tarabalka, Jon Atli Benediktsson, Jocelyn Chanussot, and James C Tilton, “Advances in spectral-spatial classification of hyperspectral images,” *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2012.
- [2] Yuliya Tarabalka, Jón Atli Benediktsson, Jocelyn Chanussot, and James C Tilton, “Multiple spectral-spatial classification approach for hyperspectral data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4122–4132, 2010.
- [3] Sicong Liu, Daniele Marinelli, Lorenzo Bruzzone, and Francesca Bovolo, “A review of change detection in multitemporal hyperspectral images: Current techniques, applications, and challenges,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 140–158, 2019.
- [4] Ashbindu Singh, “Review article digital change detection techniques using remotely-sensed data,” *International Journal of Remote Sensing*, vol. 10, no. 6, pp. 989–1003, 1989.
- [5] Francesca Bovolo and Lorenzo Bruzzone, “The time variable in data fusion: A change detection perspective,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 3, no. 3, pp. 8–26, 2015.
- [6] Jinru Xue and Baofeng Su, “Significant remote sensing vegetation indices: A review of developments and applications,” *Journal of Sensors*, vol. 2017, 2017.
- [7] R. D. Johnson and ES Kasischke, “Change vector analysis: A technique for the multispectral monitoring of land cover and condition,” *International Journal of Remote Sensing*, vol. 19, no. 3, pp. 411–426, 1998.
- [8] King-Sun Fu and JK Mui, “A survey on image segmentation,” *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, 1981.
- [9] Yun Zhang, Xuezhi Feng, and Xinghua Le, “Segmentation on multispectral remote sensing image using watershed transformation,” in *2008 Congress on Image and Signal Processing*. IEEE, 2008, vol. 4, pp. 773–777.
- [10] Yuliya Tarabalka, Jocelyn Chanussot, and Jon Atli Benediktsson, “Segmentation and classification of hyperspectral images using watershed transformation,” *Pattern Recognition*, vol. 43, no. 7, pp. 2367–2379, 2010.
- [11] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [12] Vaïa Machairas, Matthieu Faessel, David Cárdenas-Peña, Théodore Chabardes, Thomas Walter, and Etienne Deccenière, “Waterpixels,” *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3707–3716, 2015.
- [13] Jian Yao, Marko Boben, Sanja Fidler, and Raquel Urtasun, “Real-time coarse-to-fine topologically preserving segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2947–2955.
- [14] Martino Pesaresi and Jon Atli Benediktsson, “A new approach for the morphological segmentation of high-resolution satellite imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 2, pp. 309–320, 2001.
- [15] Pablo Quesada-Barriuso, Francisco Argüello, and Dora B Heras, “Spectral-spatial classification of hyperspectral images using wavelets and extended morphological profiles,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1177–1185, 2014.
- [16] Bernhard Scholkopf and Alexander J Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, Adaptive Computation and Machine Learning series, 2018.
- [17] Yushi Chen, Hanlu Jiang, Chunyang Li, Xiuping Jia, and Pedram Ghamisi, “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.
- [18] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li, “Deep convolutional neural networks for hyperspectral image classification,” *Journal of Sensors*, vol. 2015, 2015.
- [19] OpenMP Architecture Review Board, “OpenMP página web,” <https://www.openmp.org/>, [Online; último acceso el 8 de mayo de 2021].
- [20] Alberto S Garea, Dora B Heras, and Francisco Argüello, “Gpu classification of remote-sensing images using kernel elm and extended morphological profiles,” *International Journal of Remote Sensing*, vol. 37, no. 24, pp. 5918–5935, 2016.
- [21] Pablo Quesada-Barriuso, Dora Blanco Heras, and Francisco Argüello, “Gpu accelerated waterpixel algorithm for superpixel segmentation of hyperspectral images,” *The Journal of Supercomputing*, pp. 1–13, 2021.
- [22] Javier López-Fandino, Dora B Heras, and Francisco Argüello, “Multiclass change detection for multidimensional images in the presence of noise,” in *High-Performance Computing in Geoscience and Remote Sensing VIII*. International Society for Optics and Photonics, 2018, vol. 10792, p. 1079204.
- [23] Mehmet Sezgin and Bülent Sankur, “Survey over image thresholding techniques and quantitative performance evaluation,” *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–165, 2004.
- [24] Svante Wold, Kim Esbensen, and Paul Geladi, “Principal component analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.
- [25] Jon Atli Benediktsson, Martino Pesaresi, and Kolbeinn Amason, “Classification and feature extraction for remote sensing images from urban areas based on morphological transformations,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1940–1949, 2003.
- [26] Prashanth Reddy Marpu, Mattia Pedergnana, Mau-

- ro Dalla Mura, Stijn Peeters, Jon Atli Benediktsson, and Lorenzo Bruzzone, “Classification of hyperspectral data using extended attribute profiles based on supervised and unsupervised feature extraction techniques,” *International Journal of Image and Data Fusion*, vol. 3, no. 3, pp. 269–298, 2012.
- [27] John Alan Richards and JA Richards, *Remote sensing digital image analysis*, vol. 3, Springer, 1999.
 - [28] Diederik P. Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” 2017.
 - [29] Xavier Glorot and Yoshua Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Yee Whye Teh and Mike Titterington, Eds., Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010, vol. 9 of *Proceedings of Machine Learning Research*, pp. 249–256, PMLR.