

# Detección de Anomalías en Imagen Multiespectral Basada en *Isolation Forest* y Redes de Deep Learning

Javier López-Fandiño<sup>1,3</sup>, Daniel González Filgueira<sup>2</sup>, Dora B. Heras<sup>1,3</sup> y Francisco Argüello<sup>3</sup>

**Resumen**— Las imágenes multiespectrales son fundamentales en sensado remoto en tareas como la clasificación, detección de cambios o anomalías. Este trabajo propone el uso de los algoritmos *Isolation Forest* y *Deep SAD* para detección de anomalías en imágenes multiespectrales. *Isolation Forest*, basado en árboles de aislamiento, es altamente paralelizable y poco sensible a variaciones de dimensionalidad, mientras que *Deep SAD* se basa en un autoencoder convolucional para una detección más robusta, manejando mejor la variabilidad de las anomalías. Ambos algoritmos son evaluados con diferentes estrategias de cálculo de umbral óptimo, para binarizar las puntuaciones de anomalía. Los resultados experimentales muestran que ambos métodos superan al algoritmo de referencia RX, destacando la alta tasa de detección de *Isolation Forest* y el balance entre identificación de anomalías y reducción de falsos positivos de *Deep SAD*.

**Palabras clave**— Aprendizaje profundo, detección de anomalías, imagen multiespectral, *Isolation Forest*, *Deep SAD*.

## I. INTRODUCCIÓN

EN el campo científico del sensado remoto, las imágenes multiespectrales son aquellas que capturan información en distintos rangos del espectro electromagnético mediante el uso de sensores avanzados. A diferencia de los sensores de las cámaras convencionales, que solo capturan luz en el rango visible, estos sensores abarcan una amplia gama de longitudes de onda. Esta capacidad de capturar múltiples bandas del espectro electromagnético proporciona una mayor cantidad de información de utilidad que las hace idóneas para el análisis de alta precisión de la corteza terrestre.

En los últimos años, gracias a la evolución del aprendizaje automático, la investigación científica en la detección de anomalías sobre imágenes multiespectrales ha generado un gran interés, proporcionando muy buenos resultados. Para ello, se emplean diferentes técnicas y algoritmos para identificar aquellos elementos de un entorno que cuentan con una mayor diferenciación con respecto al resto de la escena [1]. En este trabajo se busca monitorizar un entorno semi-natural detectando edificaciones u otras cons-

trucciones humanas que no deberían estar presentes en dicho entorno.

La detección de anomalías consiste en dividir las muestras disponibles entre observaciones normales y anómalas. Las observaciones normales se corresponden con las aquellas cuyo espectro coincide con el de los elementos que se espera encontrar en la imagen, por ejemplo, en un entorno natural, cualquier tipo de vegetación o agua. Por otro lado, las anomalías son aquellas muestras que difieren del resto: construcciones llevadas a cabo por el ser humano, como casas o carreteras.

La detección de anomalías puede abordarse tanto con técnicas supervisadas como no supervisadas. Entre las no supervisadas la técnica de referencia en la literatura es el algoritmo Reed-Xiaoli (RX) [2], basado en el cálculo de distancias entre pares de muestras. Las técnicas supervisadas de referencia en la actualidad están mayoritariamente basadas en el uso de *Deep Learning*, como pueden ser las Redes Neuronales Convolucionales -*Convolutional Neural Network (CNN)*- [3].

El objetivo principal de este trabajo es adaptar los métodos *Isolation Forest* y *Deep SAD*, de aprendizaje no supervisado y semisupervisado, respectivamente, para su uso con imágenes multiespectrales y evaluar su rendimiento en la tarea de detección de anomalías en imágenes multiespectrales de sensado remoto respecto al algoritmo de referencia RX.

## II. TRABAJO RELACIONADO

El algoritmo no supervisado empleado de forma más recurrente en la detección de anomalías en imágenes multiespectrales, y que habitualmente se toma como *benchmark*, es el RX [2]. RX es una técnica no supervisada basada en el concepto de la distancia de Mahalanobis. Caracteriza las anomalías midiendo las desviaciones espectrales respecto a la media local de los píxeles cercanos. Una vez obtenida la puntuación de anomalía para los distintos patrones, se requiere de un punto de corte o umbral para discernir píxeles normales de anómalos. La efectividad del algoritmo RX radica en su capacidad para manejar datos de alta dimensión e identificar anomalías sutiles. Este método ya fue estudiado para la detección de construcciones humanas dentro de ecosistemas fluviales en imágenes multiespectrales, logrando resultados satisfactorios [4].

Otro conjunto de métodos se basan en el aprendi-

<sup>1</sup>Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, 15782, Santiago de Compostela, (e-mail: javier.lopez.fandino@usc.es; dora.blanco@usc.es).

<sup>2</sup>Escola Técnica Superior de Enxeñaría, Universidade de Santiago de Compostela, 15782, Santiago de Compostela, (e-mail: daniel.gonzalez.filgueira@rai.usc.es).

<sup>3</sup>Dpto. de Arquitectura y Tecnología de Computadores, Universidade de Santiago de Compostela, 15782, Santiago de Compostela, (e-mail: francisco.arguello@usc.es)

zaje supervisado. Inicialmente, para realizar la tarea de clasificación de imágenes, así como de detección de anomalías (clasificación binaria), se hizo uso de métodos tradicionales como pueden ser las Máquinas de Vectores de Soporte -*Support Vector Machines (SVM)*-. Estos algoritmos proporcionaban un buen compromiso entre rendimiento y coste computacional, resultando de utilidad en problemas convencionales. De forma sintetizada, SVM es un método de aprendizaje supervisado consistente en encontrar un hiperplano óptimo que separe los datos en diferentes clases de manera que la distancia entre el hiperplano y los puntos de datos más cercanos de cada clase sea maximizada [5]. Las SVM pueden manejar tanto problemas lineales como no lineales mediante el uso de diferentes funciones de *kernel*, lo que las hace flexibles y eficaces en una gran variedad de aplicaciones. Debido a su buen rendimiento en tareas de clasificación [5], las SVM ha sido también adaptadas en el pasado para la detección de anomalías [6].

En la literatura pueden encontrarse varios clasificadores de una sola clase desarrollados específicamente para el problema de la detección de anomalías, como puede ser la Descripción de Datos de Vectores de Soporte -*Support Vector Data Description (SVDD)*- [7]. El enfoque SVDD tiene varios beneficios respecto a las SVM. Es no paramétrico, evitando suposiciones previas sobre la distribución de los datos, y necesita menos muestras de entrenamiento, lo que ayuda a evitar el sobreajuste y generalizar mejor que otros métodos clásicos.

En los últimos años, gran parte de la investigación científica se ha centrado en soluciones de *Deep Learning*, un subcampo del aprendizaje automático que utiliza Redes Neuronales Profundas -*Deep Neural Networks (DNN)*-. La estructura de las DNNs incluye múltiples capas ocultas, lo que permite capturar en detalle patrones y características complejas de un gran volumen de datos de entrada para identificar anomalías sutiles que podrían pasar desapercibidas con métodos más simples.

En concreto, para problemas de clasificación y detección de anomalías en imágenes hiperespectrales es común utilizar CNNs[3]. Las CNNs han sido estudiadas y aplicadas tradicionalmente en el campo de la visión por computador. Se conforman básicamente de tres tipos de capas principales: convolucionales, de agrupación (*pooling*), y totalmente conectadas (*fully connected*).

La capa convolucional es la primera capa de una CNN. Las capas convolucionales pueden ir seguidas de otras capas convolucionales o de capas de *pooling*. La capa final es la capa *fully connected*, en la que se asigna la clase final. Con cada capa, la CNN aumenta en complejidad, identificando áreas mayores de la imagen. Las primeras capas se centran en características simples, como colores y bordes. A medida que los datos de la imagen avanzan a través de las capas, la CNN comienza a reconocer elementos o formas más grandes hasta que finalmente identifica el objeto esperado.

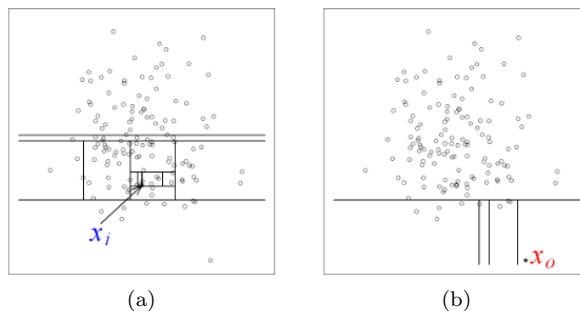


Fig. 1: Ejemplo de procesos de aislamiento de un punto normal (a) y de uno anómalo (b) [12].

#### A. Métodos de cálculo de umbral

En la tarea de detección de anomalías, resulta común la necesidad de convertir una puntuación continua de probabilidad de anomalía en una decisión binaria sobre la clase final que se asigna a cada píxel. Esta tarea, conocida como *thresholding* divide los píxeles de la imagen entre anomalías (*foreground*) y fondo (*background*). En la literatura se han descrito diferentes métodos para llevar a cabo esta tarea:

- **Otsu** [8]: selecciona el umbral que minimiza la varianza intra-clase ponderada, dividiendo las puntuaciones en dos grupos estadísticamente diferenciados.
- **Knee** [9]: detecta un cambio abrupto en la pendiente de la curva suavizada de las puntuaciones de anomalía, identificando el *punto de rodilla* como umbral óptimo.
- **ISODATA** [10]: aplica un proceso iterativo de clasificación y actualización del umbral basado en medias gaussianas hasta converger.
- **Li** [11]: busca el umbral que minimiza la entropía cruzada entre la distribución observada y una distribución de referencia, mediante un ajuste iterativo.

### III. DETECCIÓN DE ANOMALÍAS MULTIESPECTRAL BASADA EN *Isolation Forest*

*Isolation Forest* es un método no supervisado cuyo objetivo es obtener puntuaciones de anomalía contabilizando las divisiones aleatorias requeridas para que cada píxel de una imagen sea aislado del resto. No es necesario calcular distancias entre observaciones, requiriendo así menos memoria y un menor tiempo de ejecución que otros métodos [12]. El método se basa en los valores que toman los píxeles en las diferentes bandas de la imagen para caracterizar las anomalías como aquellos píxeles que requieren un menor número de divisiones aleatorias para ser aislados respecto al resto del conjunto de entrenamiento. La Figura 1 muestra un ejemplo donde se puede observar que el número de divisiones necesarias para aislar una observación normal es mayor que el necesario para aislar una observación anómala.

*Isolation Forest*, sin embargo, sí requiere el uso de un mecanismo de *thresholding* para discernir entre las dos posibles categorías a partir de las puntuaciones de anomalía de los píxeles. Una diferencia sig-

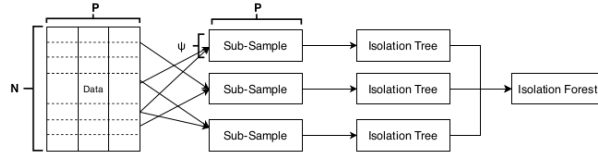


Fig. 2: Diagrama de la fase de entrenamiento de *Isolation Forest* [13]

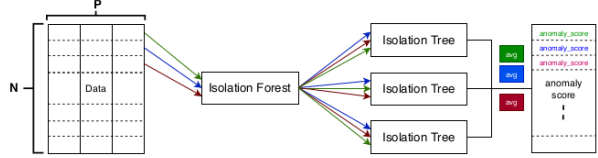


Fig. 3: Diagrama de la fase de evaluación de *Isolation Forest* [13]. Cada flecha representa distintas observaciones del conjunto de entrada.

nificativa respecto a otras alternativas, como RX, es que el modelo *Isolation Forest* se obtiene mediante la combinación de múltiples estructuras de tipo árbol, cada una entrenada con una muestra distinta generada por *bootstrapping* a partir de los datos originales. El *bootstrapping* consiste en seleccionar instancias aleatoriamente del conjunto de entrenamiento con reemplazo. La puntuación final para una observación será el promedio de divisiones necesarias para aislarla en los diferentes árboles del resto de su conjunto de entrenamiento. Valores más bajos indican mayor probabilidad de anomalía.

De forma resumida, *Isolation Forest* se basa en realizar los siguientes pasos:

- Selección aleatoria de una característica de las muestras del conjunto de datos. En el caso de imágenes multiespectrales, una banda (o subconjunto de ellas).
- Selección aleatoria de un punto de corte entre el valor mínimo y máximo de dicha característica.
- Análisis del camino generado por esta partición en el árbol. Datos anómalos darán lugar a caminos más cortos.

Aprovechando el hecho de que las anomalías generan caminos más cortos, se puede limitar la altura del árbol para así optimizar el uso de memoria del algoritmo. Además, dado que los árboles se construyen sobre submuestras del conjunto inicial mediante *bootstrapping*, la fase de entrenamiento es altamente paralelizable [13] ya que los árboles generados son independientes.

En la Figura 2 puede observarse el proceso de división del conjunto inicial en submuestras, a partir de las cuales se entrenan los distintos *Isolation Tree*, para finalmente computar el promedio para cada una de ellas, como puede observarse en la Figura 3.

*Isolation Forest* obtiene el número de divisiones requeridas para el aislamiento de las instancias del conjunto de entrada en los diferentes *Isolation Tree*, para finalmente computar el promedio para cada una de ellas, como puede observarse en la Figura 3.

Dado que el método de *Isolation Forest* es no supervisado, para su funcionamiento en ningún momento se requerirá de información etiquetada, pues las puntuaciones de anomalía para cada píxel se cal-

cularán únicamente en base a los valores de sus diferentes bandas en la imagen de entrada.

Tanto el entrenamiento del árbol como la función de decisión han sido paralelizados para dividir la carga computacional y reducir el tiempo total de ejecución.

El rendimiento del algoritmo *Isolation Forest* depende de varios hiperparámetros, cuya optimización resulta clave para mejorar la detección de anomalías y minimizar el tiempo de ejecución. En este trabajo se han analizado los siguientes:

- **Número de estimadores:** determina cuántos árboles de aislamiento se generan. Un mayor número reduce la varianza entre ejecuciones, pero incrementa el coste computacional.
- **Número de muestras:** cantidad de píxeles usados para entrenar cada árbol. Más muestras implican árboles más complejos y mayor coste computacional, pero también pueden mejorar la precisión.
- **Número de características:** número de bandas utilizado en cada división. Aumentar este valor puede acelerar el aislamiento al considerar más información por nodo, sin afectar significativamente al tiempo de ejecución.
- **Proporción esperada de anomalías:** define el porcentaje de anomalías asumido por el modelo al calcular el umbral de decisión. No influye en las puntuaciones de anomalía, pero permite utilizar un proceso de *thresholding* propio del algoritmo basado en este valor.

#### IV. Deep SAD: DETECCIÓN DE ANOMALÍAS MULTIESPECTRAL SEMI-SUPERVISADA

Hasta el momento, se han citado, por un lado, métodos supervisados, los cuales requieren un conjunto de datos etiquetados sobre los que realizar el entrenamiento, y por otro lado, métodos no supervisados, que a partir de la información de entrada extraen características y realizan diferentes cálculos para la predicción de la clase (anomalía o no anomalía) de cada píxel de la imagen de entrada. No obstante, en la práctica es común contar tan sólo con algunos datos etiquetados, de forma que no se dispone de un conjunto de datos totalmente etiquetado para poder ejecutar un método supervisado; pero al mismo tiempo, se tiene información potencialmente útil para aprender a discernir entre las clases, que en un método no supervisado no se podría aprovechar. La inclusión de esta información en un modelo nos lleva a hablar de métodos de aprendizaje semisupervisados.

El aprendizaje semisupervisado permite combinar ventajas del aprendizaje supervisado y no supervisado. Los clasificadores de aprendizaje profundo tradicionales utilizan solo datos etiquetados para entrenar. Sin embargo, las instancias etiquetadas suelen ser difíciles de obtener, ya que requieren de implicación humana y de una amplia experiencia previa. Sin embargo, los datos no etiquetados pueden ser relativamente fáciles de recopilar. El aprendizaje semisu-

pervisado aborda este problema utilizando una gran cantidad de datos no etiquetados, junto con los datos etiquetados disponibles, para construir mejores clasificadores [14].

La mayoría de clasificadores de aprendizaje profundo semisupervisados asumen que instancias similares pertenecen a la misma clase (*cluster assumption*) [14]. Esta asunción es incorrecta para la detección de anomalías, ya que estas no necesariamente serán similares entre sí. *Deep SAD* [15] consiste en un método de aprendizaje profundo semisupervisado que aprovecha la información etiquetada de datos normales y anómalos para mejorar la capacidad de discriminación. Está basado en el modelado de la distribución latente de los datos normales de forma que tenga una baja entropía y la de los datos anómalos para que tenga alta entropía. Internamente, *Deep SAD* funciona mediante una estructura de autoencoder convolucional [16]. Estas estructuras consisten en una red neuronal diseñada para aprender una representación eficiente y comprimida de los datos de entrada, compuesta por dos partes principales: el codificador (*encoder*) y el decodificador (*decoder*).

- El codificador es responsable de transformar la entrada original en una representación de menor dimensión (representación latente), capturando las características más importantes de los datos. Utiliza capas convolucionales, que aplican filtros para detectar patrones locales y reducir la dimensionalidad manteniendo la información relevante. Esta parte de la red es crucial para extraer las características clave que describen los datos de entrada.
- El decodificador, por otro lado, toma la representación comprimida y trata de reconstruir la entrada original a partir de ella. También utiliza capas convolucionales, pero en este caso, para expandir la representación comprimida y aproximarse lo máximo posible a los datos de entrada originales. La diferencia entre la entrada original y la reconstruida proporciona una medida del error de reconstrucción, que es utilizado para ajustar los pesos de la red durante el entrenamiento.

En lo referente a los datos etiquetados disponibles, *Deep SAD* trata estos de forma diferenciada en función de si corresponden a píxeles considerados como anomalía o no. Para los datos normales etiquetados se hace uso, de igual forma que para SVDD [7], de una función de error distancia cuadrática al centro de una hipersfera. Para las anomalías, se penaliza en base al inverso de la distancia, ya que, siguiendo la suposición de que las anomalías no tienen que estar agrupadas, se espera que estas estarán ubicadas más lejos del centro. La puntuación de anomalías vendrá por tanto dada por la distancia al centro. Nuevamente, se deberá aplicar un método de *thresholding* para obtener la decisión final a partir de dichas puntuaciones.

Para el funcionamiento del algoritmo *Deep SAD*,

además de los parámetros habituales en las redes de este tipo, como el número de épocas, tamaño de *batch*, la tasa de aprendizaje o el *weight decay*, hay una serie de parámetros propios a tener en cuenta:

- **Ratio de datos normales conocidos.** Define la proporción de muestras normales conocidas (etiquetadas) en el conjunto de datos. Es crucial para garantizar que una parte representativa de los datos normales esté correctamente etiquetada, facilitando así el aprendizaje del modelo.
- **Ratio de anomalías conocidas.** Determina la proporción de muestras anómalas conocidas (etiquetadas) en el conjunto de datos. Tener una cantidad suficiente de anomalías etiquetadas ayuda al modelo a identificar y aprender las características distintivas de las anomalías.
- **Ratio de contaminación.** Especifica la proporción de muestras anómalas no etiquetadas (desconocidas) dentro de los datos no etiquetados. Es esencial para simular condiciones del mundo real donde las anomalías no siempre están etiquetadas, lo que permite al modelo desarrollar habilidades de detección de anomalías en un entorno más realista.

A partir de los parámetros previamente introducidos y de las etiquetas del conjunto de entrenamiento se realiza una serie de procesamientos para obtener la configuración semisupervisada. *Deep SAD* permite detectar varios tipos de anomalías. No obstante, este trabajo siempre se considerará una única clase de anomalías.

En primer lugar, se seleccionan los índices (píxeles) correspondientes a cada tipo de muestra. Se resuelve un sistema de ecuaciones lineales para determinar el número de muestras en cada categoría (conocidas y no etiquetadas). Este sistema de ecuaciones ayuda a calcular cuántas muestras normales conocidas, muestras normales no etiquetadas, muestras anómalas no etiquetadas y muestras anómalas conocidas son necesarias para cumplir con las proporciones especificadas por los parámetros. Una vez determinadas las cantidades necesarias, se realiza un muestreo aleatorio de índices para cada categoría basada en las cantidades calculadas. Así, se seleccionan aleatoriamente los índices de las muestras que serán incluidas en cada categoría.

Posteriormente, se generan las etiquetas originales y las etiquetas semisupervisadas para las muestras seleccionadas. Finalmente, se obtiene una lista con los índices de las muestras seleccionadas y otra con las etiquetas semisupervisadas de estas muestras. Para los índices obtenidos, se asignan sus respectivas etiquetas semisupervisadas en el conjunto de entrenamiento inicial, de forma que finalmente sólo se mantiene este subconjunto de índices del total que había previamente en el conjunto de entrenamiento.

En los últimos años, el uso de *autoencoders* para el preentrenamiento de modelos de *machine learning* es una práctica común que ha demostrado tener una serie de ventajas [17]. Principalmente, permite mejo-



rar la inicialización de los pesos de las neuronas, de forma que el entrenamiento posterior sea más eficiente y converja más rápido. También resulta de ayuda para evitar problemas frecuentes en el entrenamiento como el desvanecimiento de gradientes.

En esencia, el preentrenamiento de *Deep SAD* consiste en el uso de un autoencoder no supervisado que utiliza todos los datos proporcionados, cuyos pesos de convergencia se emplean para la inicialización de los pesos de la red final.

A continuación, se detalla la arquitectura de capas considerada en esta implementación para construir dicho autoencoder:

- El *encoder* consiste en tres capas convolucionales, cada una seguida por una capa de normalización por lotes y una capa de *maxpooling*. Finalmente, una capa totalmente conectada reduce las características extraídas a una representación final de tamaño 128. Las capas convolucionales permiten la detección de patrones y características importantes, mientras que la normalización por lotes estabiliza y acelera el proceso de entrenamiento. La capa de *maxpooling* reduce la dimensionalidad, permitiendo una mayor generalización eficiencia computacional. La capa totalmente conectada final reduce las características extraídas a una representación compacta que encapsula la información más relevante de la imagen.
- El *decoder* cuenta con cuatro capas de convolución transpuesta, inicializando sus pesos de acuerdo con el método Xavier [18], seguidas cada una de ellas de una capa de normalización por lotes. Las capas de convolución transpuesta permiten aumentar progresivamente la resolución de las características aprendidas y reconstruir la imagen original. El método Xavier utilizado para la inicialización distribuye los pesos de manera uniforme en un rango específico, optimizando el flujo de gradientes durante el entrenamiento y mejorando la convergencia de la red.

La red de entrenamiento tiene la misma estructura que el encoder descrito para el preentrenamiento. No obstante, a diferencia de el preentrenamiento, donde sólo se tiene la información de los datos asociados a los diferentes píxeles de los parches, en el entrenamiento también se hace uso de las etiquetas semisupervisadas para aprender a clasificar los índices de la imagen.

Mediante el entrenamiento, se busca transformar los datos a un espacio comprimido de menor dimensión, en el que se medirá la distancia de los diferentes puntos u observaciones al centro de una hiperesfera. Así, en este caso, las puntuaciones de anomalía se determinan como las distancias de las diferentes observaciones al centro de dicha hiperesfera en el espacio comprimido.

Como en el caso de *Isolation Forest*, se requiere de la aplicación de un método de *thresholding* para ob-

tener las clases predichas por el algoritmo para los diferentes índices de la imagen de entrada. A partir de las puntuaciones asignadas a cada pixel se computa el valor del umbral utilizando el método especificado.

## V. RESULTADOS EXPERIMENTALES

### A. Hardware

Para los experimentos realizados en este trabajo se ha utilizado un equipo con un procesador AMD Ryzen 7 5800H, 16 GB de RAM y tarjeta gráfica NVIDIA GeForce RTX 3060 Mobile con 6 GB de VRAM GDDR6. Los experimentos se han ejecutado en la versión de kernel 6.5.0-35-generic del SO Ubuntu 22.04.4 LTS, con Python 3.10.12, CUDA 12.2 y PyTorch 2.2.2.

Para evaluar el rendimiento de los métodos propuestos se utilizaron las métricas habituales en la tarea de detección de anomalías: número de verdaderos positivos (VP), que corresponde a los píxeles anómalos que han sido correctamente clasificados, número de falsos positivos (FP), que contabiliza los píxeles que no representan anomalías pero han sido clasificados erróneamente como tales, el número de verdaderos negativos (VN) como la cantidad de píxeles de fondo correctamente identificados y el número de falsos negativos (FN) como aquellos píxeles que han sido clasificados de forma incorrecta como fondo. Se incluye también el porcentaje de anomalías detectadas, que refleja la sensibilidad del modelo, y el porcentaje del fondo detectado, que mide su especificidad. Además, se emplea el área bajo la curva ROC (AUC) como indicador global de rendimiento, ya que cuantifica la capacidad del modelo para discriminar entre anomalías y fondo, siendo su valor máximo uno.

### B. Conjunto de datos

La Tabla I resume las principales características de las dos imágenes multiespectrales de 5 bandas consideradas, Zona 1 y Zona 2, correspondientes a imágenes capturadas en un entorno semi-natural cercano al Río Oitavén, en Galicia. Los datos han sido adquiridos con el sensor multiespectral MicaSense RedEdge montado en un Vehículo Aéreo No Tripulado (UAV) especializado. Este sensor captura imágenes en cinco bandas espectrales: azul (475 nm), verde (560 nm), rojo (668 nm), *red-edge* (717 nm) e infrarrojo cercano (NIR) (840 nm). Las imágenes se recopilieron durante el verano de 2018 desde una altitud de 120 metros, logrando una alta resolución espacial de 8,2 cm por píxel [19]. Como se puede ver, cada imagen presenta alrededor del 4 % de anomalías sobre el total de píxeles. El tamaño de las imágenes corresponde al caso donde la información de los píxeles se almacena en un formato de 4 bytes.

En la Figura 4, se muestra una composición RGB de las imágenes consideradas, junto con la información de referencia de anomalías disponible, siendo las anomalías marcadas en rojo.

Nombre	Zona 1	Zona 2
Anchura	3807	2081
Altura	2141	957
Bandas espectrales	5	5
Anomalías (píxeles)	321,710	83,130
No anomalías (píxeles)	7,829,077	1,908,387
Anomalías (%)	3.95	4.17
No anomalías (%)	96.05	95.83
Tamaño (MB)	155.5	38

Tabla I: Descripción del dataset Río Oitavén, compuesto por dos imágenes multiespectrales.

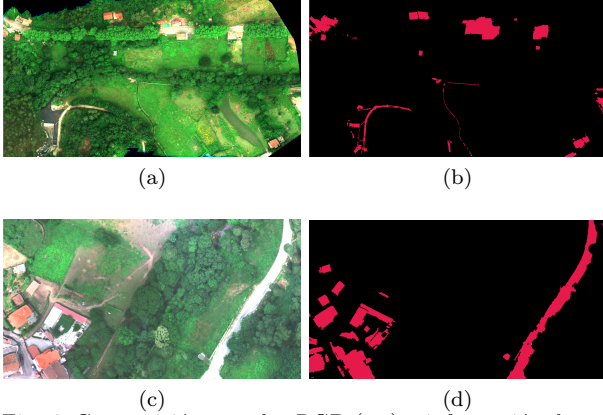


Fig. 4: Composición en color RGB (a,c) e información de referencia de anomalías (b,d) para Zona 1 y Zona 2, respectivamente. Anomalías marcadas en rojo.

### C. Resultados

La parametrización para cada método e imagen incluído en estos resultados se ha obtenido mediante una batería experimental, obteniendo el resultado promedio de 5 ejecuciones para cada combinación de parámetros y buscando maximizar la capacidad de detección de anomalías. Para *Isolation Forest* se han utilizado 40 estimadores, 4 características, 4000 muestras y el *threshold* interno de *Isolation Forest*, indicando una contaminación de 0.25 para Zona 1 y los mismos parámetros incrementando a 300 el número de estimadores para Zona 2. El uso de un mayor número de estimadores en la imagen más pequeña está directamente relacionado con el coste computacional de incrementar este parámetro en la imagen cuyo tamaño es mayor. El método de *threshold* interno basado en la contaminación ha obtenido sistemáticamente resultados mejores que los métodos alternativos como Otsu o Kneec.

En el caso de *Deep SAD*, se entrena el modelo durante 10 épocas, utilizando una tasa de aprendizaje de 0.0001 y *weight decay* de  $1 \times 10^{-6}$ . Los mejores resultados se obtuvieron con el *threshold* Otsu para Zona 1 y Li para Zona 2. Para el entrenamiento etiquetado de *Deep SAD* se seleccionan el mismo número de muestras para las clases anomalía y no anomalía, asegurando así que se toman suficientes muestras para entrenar la clase anomalía a pesar del gran desbalanceo entre el número de muestras inherente a este problema. Se utilizan como muestras etiquetadas el 50 % de las anomalías disponibles.

A partir de los índices de entrenamiento, se procede con la extracción de parches de la imagen. El tamaño de *patch* adecuado dependerá principalmen-

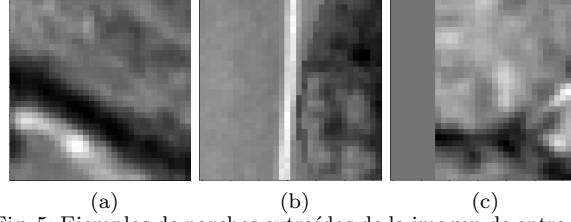


Fig. 5: Ejemplos de parches extraídos de la imagen de entrada. Parches correspondientes a píxeles en el interior de la imagen (5a,5b). Parche correspondiente a un píxel cercano al borde de la imagen (5c).

te de la dimensión espacial de la imagen. Un tamaño muy grande supondría un consumo de memoria demasiado elevado. Se ha seleccionado experimentalmente un tamaño de  $32 \times 32$ . Además, para aquellos píxeles de los que no es posible extraer un *patch* centrado en ellos, debido a su cercanía a algún margen de la imagen, se aplica un *padding* con los valores de las diferentes bandas del valor central del parche, como se aprecia en la Figura 5c.

En la Figura 6 se muestran los mapas de detección de anomalías obtenidos para Zona 1 y Zona 2. Puede observarse que ambos métodos propuestos mejoran la capacidad de detección de anomalías de RX. Como contrapartida, el número de falsos positivos es más elevado, especialmente en el caso de Zona 1.

En la Tabla II se resumen los resultados obtenidos por los algoritmos *Isolation Forest* y *Deep SAD* sobre los conjuntos de prueba Zona 1 y Zona 2. Se incluyen los resultados del método RX [4] como referencia. Como se puede apreciar, los dos métodos propuestos en este trabajo logran mejorar el porcentaje de anomalías detectadas respecto a RX en ambas imágenes. Esto está alineado con el objetivo principal de este trabajo, detectar el mayor número de anomalías posible.

El método *Isolation Forest* es el que mejor capacidad de detección ofrece, obteniendo valores de AUC cercanos a 0,9 en ambas imágenes. Además, es el algoritmo que permite detectar un mayor porcentaje de anomalías, superando en ambos conjuntos de prueba el 90 % de detección de las mismas, manteniendo un nivel de falsos positivos relativamente bajo.

Por su parte, *Deep SAD* también logra mejores resultados de detección global que RX (mayor AUC). No obstante, no logra llegar a los porcentajes de detección de anomalías logrados por *Isolation Forest*. Sin embargo, consigue un número de falsos positivos inferior a este. Al tratarse *Deep SAD* de una red de *Deep Learning*, ofrece también otro tipo de ventajas, como el hecho de poder ser reentrenada con nuevos datos para adaptarse a cambios en el entorno o en las características de las anomalías.

## VI. CONCLUSIONES

En este trabajo se han adaptado y evaluado los métodos *Isolation Forest* y *Deep SAD* para la detección de anomalías en imágenes multiespectrales. Los resultados obtenidos han demostrado la viabilidad y

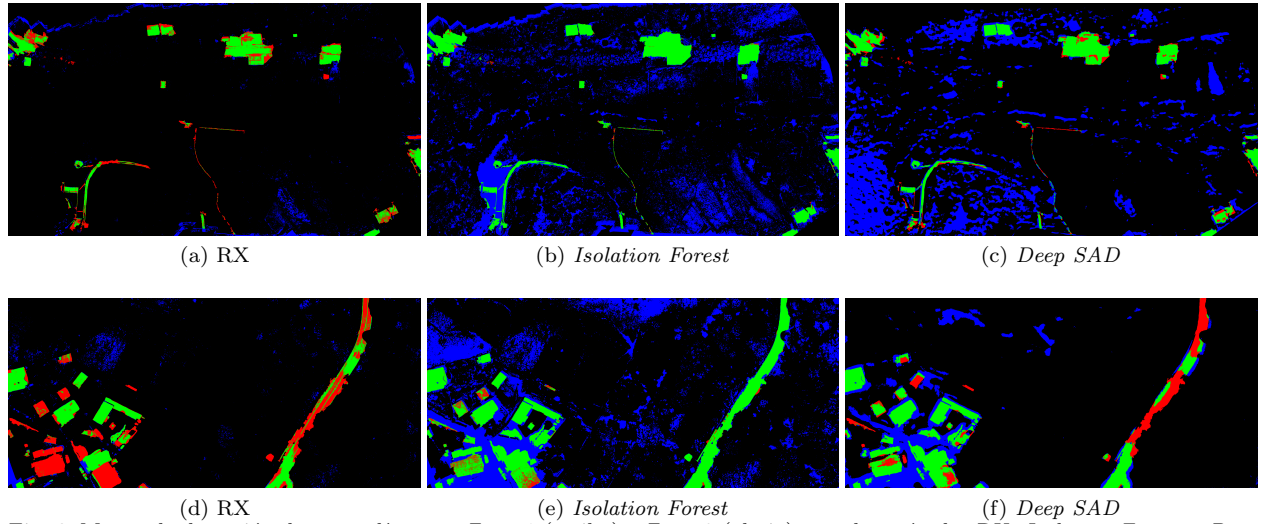


Fig. 6: Mapas de detección de anomalías para Zona 1 (arriba) y Zona 2 (abajo) para los métodos RX, *Isolation Forest* y *Deep SAD*. El color verde representa VP, el color azul FP, el color rojo FN y el color negro VN.

Imagen	Método	AUC	Anomalías (%)	Fondo (%)	VP	VN	FP	FN	Tiempo (s)
Zona 1	RX	0,818	65,464	<b>98,124</b>	210605	<b>7255470</b>	<b>138711</b>	111105	<b>16,239</b>
	<i>Isolation Forest</i>	<b>0,898</b>	<b>97,090</b>	82,550	<b>312356</b>	6103727	1290446	<b>9354</b>	75,653
	<i>Deep SAD</i>	0,830	81,250	85,080	261395	6661193	1167884	60315	560,107
Zona 2	RX	0,722	46,349	<b>98,033</b>	78104	<b>1786809</b>	<b>35857</b>	90407	<b>4,011</b>
	<i>Isolation Forest</i>	<b>0,872</b>	<b>93,145</b>	81,314	<b>156960</b>	1482086	340580	<b>11551</b>	132,060
	<i>Deep SAD</i>	0,816	70,740	92,310	119202	1682785	140221	49309	290,624

Tabla II: Resultados conseguidos mediante RX, *Isolation Forest* y *Deep SAD* para las imágenes multispectrales Zona 1 y Zona 2. Se muestran en negrita los mejores valores para cada métrica.

efectividad de estos enfoques en comparación con el método RX, ampliamente utilizado en el ámbito de la detección de anomalías en imágenes de teledetección.

En particular, *Isolation Forest* ha mostrado una capacidad superior para identificar anomalías en comparación con RX, logrando detectar un mayor porcentaje de píxeles anómalos. Su estrategia basada en la construcción de árboles de aislamiento le permite mejorar significativamente la detección en comparación con métodos tradicionales basados en distancias.

Por otro lado, *Deep SAD* ha demostrado ser un enfoque robusto para la detección de anomalías, con una mayor capacidad para distinguir entre clases y un mejor equilibrio entre verdaderos positivos y falsos positivos en comparación con RX.

A partir de los resultados obtenidos consideramos las siguientes vías de trabajo futuro: Por un lado, la integración de *Isolation Forest* con técnicas de *deep learning*, formando un modelo híbrido que combine las fortalezas de ambos métodos con el fin de obtener una mejor detección de anomalías. Por otro lado, incrementar la batería experimental, ampliando el estudio a una mayor cantidad de imágenes y utilizando técnicas avanzadas de ajuste de parámetros de manera sistemática.

#### AGRADECIMIENTOS

El presente trabajo ha sido financiado mediante los proyectos PID2022-141623NB-I00 y TED2021-130367B-I00, financiados por CI-

N/AEI/10.13039/501100011033 y por la Unión Europea a través del programa NextGenerationEU/PRTR. Asimismo, ha contado con financiación de la Xunta de Galicia – Consellería de Educación, Ciencia, Universidades e Formación Profesional, a través del programa de acreditación de Centros de Investigación de Galicia (2024-2027, ED431G-2023/04) y del Grupo Consolidado de Investigación (ED431C-2022/16), con cofinanciación del Fondo Europeo de Desarrollo Regional (FEDER/UE).

#### REFERENCIAS

- [1] H. Su, Z. Wu, H. Zhang, and Q. Du, "Hyperspectral anomaly detection: a survey," *IEEE Geosci Remote Sens Mag*, vol. 10, no. 1, pp. 64–90, 2021.
- [2] I. S. Reed and X. Yu, "Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1760–1770, 1990.
- [3] Wei Li, Guodong Wu, and Qian Du, "Transferred deep learning for anomaly detection in hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, 2017.
- [4] Javier López-Fandiño, Dora B. Heras, and Francisco Argüello, "Using heterogeneous computing and edge computing to accelerate anomaly detection in remotely sensed multispectral images," *The Journal of Supercomputing*, 2024.
- [5] Farid Melgani and Lorenzo Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.
- [6] Amit Banerjee, Philippe Burlina, and Chris Diehl, "A support vector method for anomaly detection in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 8, pp. 2282–2291, 2006.
- [7] A. Banerjee, P. Burlina, and C. Diehl, "A support vector method for anomaly detection in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 8, pp. 2282–2291, 2006.

- [8] Jamileh Yousefi, "Image binarization using otsu thresholding algorithm," 2011.
- [9] Mário Antunes, Joana Ribeiro, Diogo Gomes, and Rui L. Aguiar, "Knee/elbow point estimation through thresholding," in *2018 IEEE 6th International Conference on Future Internet of Things and Cloud*, 2018.
- [10] A. El-Zaart, "Images thresholding using isodata technique with gamma distribution," *Representation, Processing, Analysis, and Understanding of Images*, vol. 20, pp. 29–41, 2010.
- [11] C.H. Li and C.K. Lee, "Minimum cross entropy thresholding," *Pattern Recognition*, vol. 26, no. 4, pp. 617–625, 1993.
- [12] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou, "Isolation forest," in *IEEE International Conference on Data Mining*, 2008.
- [13] Adam Valenta and Veronika Maurerová, "Anomaly detection using extended isolation forest," *Faculty of Information Technology CTU in Prague*, 2019.
- [14] Xiaojin Zhu, "Semi-supervised learning literature survey," *Computer Sciences TR 1530*, 2005.
- [15] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft, "Deep semi-supervised anomaly detection," in *International Conference on Learning Representations (ICLR)*, 2020.
- [16] Arian Azarang, Hafez E. Manoochehri, and Nasser Keltarnavaz, "Convolutional autoencoder-based multispectral image fusion," *IEEE Access*, vol. 7, pp. 35673–35683, 2020.
- [17] Luca Pasa and Alessandro Sperduti, "Pre-training of recurrent neural networks via linear autoencoders," in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 2014.
- [18] 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, PMLR 9:249-256*, 2010.
- [19] Francisco Argüello, Dora B Heras, Alberto S Garea, and Pablo Quesada-Barriuso, "Watershed monitoring in galicia from uav multispectral imagery using advanced texture methods," *Remote Sensing*, vol. 13, no. 14, pp. 2687, 2021.