Leveraging convolutional and graph networks for an unsupervised remote sensing labelling tool

Tulsi Patel, Mark W. Jones & Thomas Redfern

View supplementary material

Published online: 03 Feb 2026.

Submit your article to this journal

View related articles

View Crossmark data

Leveraging convolutional and graph networks for an unsupervised remote sensing labelling tool

Tulsi Patel[a], Mark W. Jones[a] and Thomas Redfern[b]

[a]Department of Computer Science, Swansea University, Swansea, UK; [b]UK Hydrographic Office, Taunton, UK

ABSTRACT

Machine learning for remote sensing imaging relies on up-to-date and accurate labels for model training and testing. Labelling remote sensing imagery is time and cost intensive, requiring expert analysis. Previous labelling tools rely on pre-labelled data for training in order to label new unseen data. In this work, we define an unsupervised pipeline for finding and labelling geographical areas of similar context and content within Sentinel-2 satellite imagery. Our approach removes limitations of previous methods by utilizing segmentation with convolutional and graph neural networks to encode a more robust feature space for image comparison. Unlike previous approaches we segment the image into homogeneous regions of pixels that are grouped based on colour and spatial similarity. Graph neural networks are used to aggregate information about the surrounding segments enabling the feature representation to encode the local neighbourhood whilst preserving its own local information. This reduces outliers in the labelling tool, allows users to label at a granular level, and allows a rotationally invariant semantic relationship at the image level to be formed within the encoding space. Our pipeline achieves high contextual consistency, with similarity scores of SSIM = 0.96 and SAM = 0.21 under context-aware evaluation, demonstrating robust organization of the feature space for interactive labelling.

1. Introduction

Rapid advancements in satellite imagery and Remote Sensing (RS) have led to an explosion of data, with the Sentinel-2 mission producing 1.6 terabytes of data daily. This wealth of data enables applications across agriculture, land use mapping, environmental monitoring, coastal management and monitoring and geological studies (Ahmadi and Pekkan 2021; Karthikeyan, Chawla, and Mishra 2020; Seale et al. 2022; Weiss, Jacob, and Duveiller 2020; L. Zhang and Zhang 2022) – all of which require high-quality labelled datasets. However, the relatively large pixel resolutions of Sentinal-2 (covering 10 to 60 m) create challenges for labelling. A single pixel, representing a 10 m by 10 m area, may cover both land and water, introducing ambiguity and mixed spectral responses. Traditional labelling approaches, whether bounding boxes, pixel-level masks, or small image segments, are time consuming and costly, requiring expert input.

1.1. Existing labelling tools

Several tools have been developed to address these challenges. For example, Li et al. introduced LabelRS ([J. Li et al. 2021]), which leverages OpenStreetMaps, Land Use Land Cover (LULC) datasets, and ArcGIS annotations to generate semantic, object, or image classification datasets. While effective, such tools rely on predefined label categories, limiting flexibility and adaptability to new or unseen classes. Beyond LabelRS, most existing solutions are not full labelling platforms but retrieval-based systems that assist annotation by retrieving and presenting similar images to a target image. These include CBIR-driven frameworks (Hou et al. 2019; Qi et al. 2020) and deep learning extensions such as NetVLAD and SatResNet-50 (Napoletano 2018), which improve retrieval accuracy but still depend on prior labels for supervised training. More recently, Patel et al. (Patel, Jones, and Redfern 2023) proposed an interactive labelling tool based on manifold embeddings
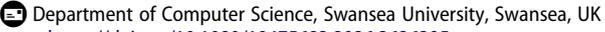
(UMAP and t-SNE), enabling experts to explore and annotate clusters in a 2D space. However, this approach struggles with outliers and lacks contextual awareness, highlighting the need for more robust, context-sensitive labelling solutions.

1.2. Methods used in these tools

To improve labelling efficiency, many approaches borrow from Content-Based Image Retrieval (CBIR), which retrieves similar content through three main steps: feature representation, indexing, and similarity measurement (Y. Li, Ma, and Zhang 2021). CBIR-based systems aim to present visually similar images to assist annotation, assuming that retrieved content shares sufficient semantic similarity.

1.2.1. Traditional feature extraction

Early CBIR methods relied on low-level descriptors capturing texture, colour, and shape characteristics. Common techniques include Scale-Invariant Feature Transform (SIFT) (Chen et al. 2020; Zhu et al. 2016), Gray-Level Co-occurrence Matrices (GLCM), wavelets, Gabor filters, and Local Binary Patterns (LBP) (X. Huang, Liu, and Zhang 2014; L. Huang et al. 2016; Ranchin and Wald 1993; Yang and Newsam 2008). These techniques primarily focus on either texture features (which capture spectral relationships) or the spectra themselves. Once extracted, these descriptors are often aggregated into mid-level representations using algorithms such as Bag of Visual Words (BoW) and Vector of Locally Aggregated Descriptors (VLAD) (Cao et al. 2010; Jégou et al. 2010; Sivic and Zisserman 2003). BoW employs K-means clustering to create a visual codebook and histograms of local features, while VLAD improves upon BoW by encoding spatial relationships between features and cluster centres. Despite their popularity, these methods typically lose semantic relationships between image features, limiting their effectiveness for complex remote sensing imagery.

1.2.2. Deep learning-based retrieval

Deep Convolutional Neural Networks (DCNNs) have transformed image retrieval by learning hierarchical features that outperform handcrafted descriptors (Imbriaco et al. 2019; Song et al. 2019). Architectures such as VGG, ResNet, OverFeat, and CaffeNet have been widely adopted (G. Wang et al. 2017), with extensions like NetVLAD and SatResNet-50 achieving state-of-the-art performance in remote sensing retrieval tasks (Napoletano 2018). These models aggregate convolutional features into compact descriptors, often combined with VLAD encoding for improved spatial awareness (G. Wang et al. 2017). Unsupervised variants using autoencoders, generative models, and hybrid approaches with SVMs or visual codebooks further reduce reliance on labelled data (Romero, Gatta, and Camps-Valls 2015; Q. Sun and Bourennane 2020; Tang et al. 2018). However, even deep models can struggle with contextual relationships when features are treated independently.

1.2.3. Graph-based and context-aware methods

To address the semantic gap, recent work incorporates Graph Neural Networks (GNNs) to encode spatial context and multi-label co-occurrence relationships. For example, Sumbul and Demir (Sumbul and Demir 2021) combine CNN feature extraction with GNN-based similarity refinement using multiple Siamese networks and triplet loss. Wang et al. (Y. Wang et al. 2021) propose a deep hashing approach using Graph Neural Networks (GNNs), where deep features are extracted from a pre-trained ResNet network. Other studies leverage GNNs for cross-modal matching (e.g. text-image retrieval) and multi-label image retrieval, demonstrating improvements when attention mechanisms are introduced (Kang et al. 2021; Yu et al. 2022). Contrastive learning and triplet loss have also been applied to enforce neighbourhood consistency in embedding spaces (Draganov and Dohn 2023; Yan et al. 2020; M. Zhang et al. 2021). These approaches highlight the importance of preserving local context, such as adjacency to water or urban density, when organizing feature spaces for retrieval and labelling. Chaudhuri et al. identified a gap between low-level features and high-level semantic concepts (B. Chaudhuri et al. 2018). They proposed a sub-graph matching strategy that emphasizes neighbourhood contexts within segmented images. While effective, this approach is computationally expensive (B. Chaudhuri et al. 2018). More recent work (U. Chaudhuri et al. 2022) improves on this by using contrastive loss between Siamese Graph Convolutional Networks (GCNs), replacing sub-graph matching. This method applies context-based attention to nodes, enhancing image retrieval performance. The key advantage of GCNs is their ability to encode neighbourhood information into each segment through message-passing operations, facilitating more context-aware feature extraction.

1.2.4. Manifold embedding and interactive labelling

Dimensionality reduction techniques such as t-SNE and UMAP have been explored to visualize high-dimensional feature spaces for interactive labelling. Halladin-Da̧browska et al. (Halladin-Dabrowska, Kania, and Kopeć 2019) applied t-SNE to enhance reference data for heterogeneous vegetation. Their method provides contextual feedback about the relationships between features, such as distinguishing urban areas from surrounding land or dense urban zones. This contextual discrimination, based on geographical proximity, allows labellers to customize datasets more effectively, ensuring they can balance and fine-tune their labelled datasets according to specific requirements. Patel et al. (Patel, Jones, and Redfern 2023) proposed an interactive tool that embeds features into a 2D space for exploration and annotation. These techniques are useful for preserving local distances between data points when reducing dimensionality, which aids in visualizing machine learning results (Ali et al. 2019; Ali, Borgo, and Jones 2021). Their approach enables experts to label clusters visually and interactively but suffers from outliers and orientation sensitivity, where strong texture gradients (e.g. land-water boundaries) distort clustering. These limitations underscore the need for embeddings that are both context-aware and rotationally invariant.

Recent work by Sun et al. (Y. Sun, Lei, Li, et al. 2025) and Sun et al. (Y. Sun, Lei, Guan, et al. 2025) introduces highly efficient frameworks for unsupervised change detection in heterogeneous and multimodal remote sensing imagery. Y. Sun, Lei, Li, et al. (2025) leverage rule-based constraints within superpixels to detect changes without explicit image comparison, while LPEM (Y. Sun, Lei, Guan, et al. 2025) preserves locality and spatial continuity through an energy-based formulation, avoiding intermediate difference images. These are highly efficient and improve robustness to imaging conditions by avoiding complex transformations for aligning heterogeneous data. Both methods share our goal of reducing reliance on predefined labels and exploiting spatial context. In contrast, our approach targets unsupervised labelling and clustering rather than change detection, learning context-aware embeddings via CNN and GNN to enable interactive exploration and semantic organization of large datasets.

1.3. Limitations of existing approaches

Overall, these studies demonstrate a progression from low-level descriptors to deep learning and graph-based approaches, with increasing emphasis on contextual relationships and interactive labelling. However, existing methods still face challenges in semantic organization, robustness to outliers, and adaptability to new classes – motivating our proposed pipeline that integrates CNNs and GNNs for context-aware, unsupervised labelling.

Existing methods face key limitations: (i) dependence on predefined label categories, reducing adaptability; (ii) loss of semantic relationships in codebook-based approaches; (iii) sensitivity to outliers and orientation, which can distort clustering; and (iv) insufficient encoding of spatial context, which is critical for remote sensing semantics. These gaps motivate the need for a more robust, context-aware solution that supports unsupervised labelling and interactive exploration.

1.4. Our contribution

To address the challenge of costly and time-consuming labelling in remote sensing, we sought to develop a toolchain that enhances efficiency by presenting experts with visually and contextually similar content for rapid annotation. Inspired by Content-Based Image Retrieval (CBIR), which retrieves similar images through feature representation, indexing, and similarity measurement (Y. Li, Ma, and Zhang 2021), we extend this concept to an unsupervised setting where semantic organization is achieved without predefined labels. Unlike prior approaches that rely on low-level descriptors or purely deep learning-based embeddings, our method integrates Convolutional Neural Networks (CNNs) for rich feature extraction and Graph Neural Networks (GNNs) for encoding neighbourhood relationships, enabling context-aware embeddings that preserve spatial semantics. This design mitigates the impact of outliers, introduces rotational invariance, and supports interactive exploration of large datasets. In contrast to existing manifold embedding tools (Patel, Jones, and Redfern 2023), which suffer from feature misalignment and orientation sensitivity, our pipeline provides a more robust and reliable foundation for visual labelling.

Specifically, we make the following contributions:

(a) We improve the coherence of embedding spaces by achieving tighter cluster formation, facilitating easier navigation through unsupervised clustering.
(b) By integrating GNNs and feature encoding on image segmentations, we demonstrate that the resulting two-dimensional embedding space is robust to spatial transformations and encodes local context.
(c) Finally, we apply this methodology to develop an advanced interactive tool for exploration and labelling, enhancing both efficiency and accuracy in dataset creation.

2. Materials and methods

The proposed pipeline (Figure 1) consists of three key components: a U-Net for feature extraction, a Graph Neural Network (GNN) for generating relational embeddings, and a graph-matching algorithm for assessing graph similarity. This design explicitly bridges the gap between low-level pixel features and high-level semantic concepts as identified by Chaudhuri et al. (2018). First, SLIC segmentation operates directly on raw pixel values to partition the image into super-pixels, creating spatially coherent regions that serve as graph nodes. These segments preserve local spectral and textural characteristics while introducing a structural representation of the image. Next, the U-Net extracts rich activation maps from the original image, which are aggregated within each segment to form node-level feature vectors. The GNN then propagates information across these nodes, encoding neighbourhood relationships such as adjacency to water or urban density. Through this message-passing process, isolated low-level descriptors are transformed into context-aware embeddings that capture both local detail and spatial semantics. Finally, graph matching and UMAP projection organize these enriched embeddings into a manifold where clusters correspond to meaningful categories (e.g. vegetation, urban, coastal), enabling interactive labelling at both chip and segment levels. This progression from raw pixels to graph-based context ensures that local pixel information is contextualized within its environment, achieving the bridging between low-level and high-level representations.

2.1. Dataset

We utilize the Sentinel-2 Water Edges Dataset (SWED) developed by the UK Hydrographic Office (Seale et al. 2022). It consists of globally distributed Sentinel-2 scenes captured 2017–2021 annotated for coastline extraction, although we do not use the labels. The images are tiles, regions of 10,980 m by 10,980 m, taken in a single pass of the satellite. Minimal cloud coverage is present within the dataset, and each tile contains a diverse set of coastal landscape features from globally distributed locations.
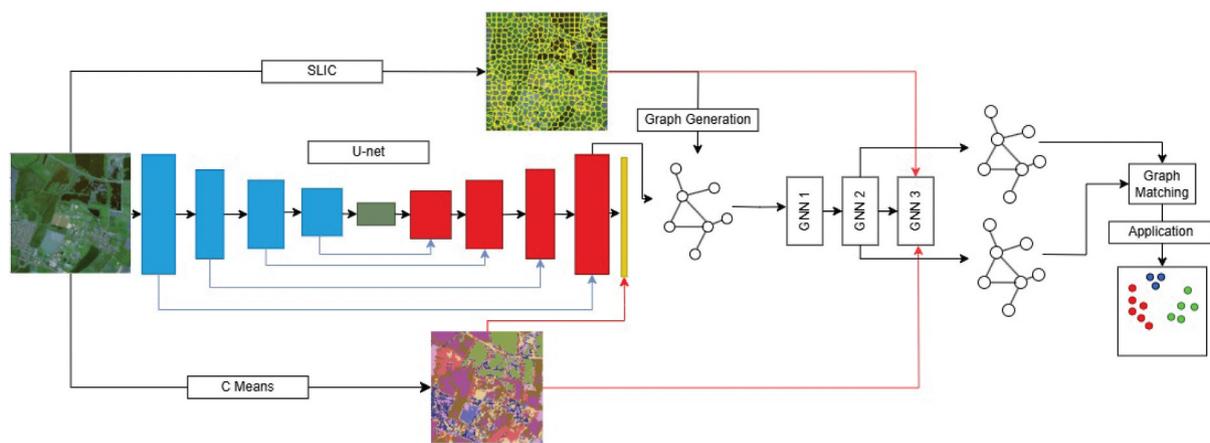

Figure 1. This diagram illustrates the data flow within the pipeline. Blue boxes represent the contracting layers of the U-Net, red boxes the expansive layers, and yellow the predictive layer. Blue lines indicate skip connections. Red lines show how data is used in the loss function, where the third GNN layer reshapes its input to enable comparison with the targets.

2.2. Fuzzy C-Means target extraction

Due to the spatial resolution of most satellite platforms, pixel assignment can be ambiguous or uncertain (Y. Zhang et al. 2024). Fuzzy C-means extraction, unlike its K-means counterpart, assigns to each data point a degree of membership to each cluster, transforming the problem into a multi-label classification. We use all 12 spectral channels for clustering. To reduce memory requirements during model training, we apply uniform subsampling by selecting every 56th pixel across the image, ensuring representative coverage while significantly lowering computational cost. After training, the clustering model is applied to the full-resolution image to obtain tile-wide classifications. The resultant centroids do not correspond to specific physical materials (e.g. water or concrete) but rather represent dominant spectral patterns, combinations of reflectance values across bands, that capture the most common pixel properties in the scene. These properties reflect typical spectral signatures rather than discrete land-cover classes, enabling soft classification and generalization for pixels that do not fully belong to any single cluster.

2.3. Pre-Proccessing

For processing by the U-Net and GNN we split the tile into chips. A chip is created by applying a sliding window of size 256×256 pixels to the larger image, with no overlap between consecutive windows. This results in 42×42 chips containing $10,752 \times 10,752$ of the original tiles $10,980 \times 10,980$ pixels. For training and testing, we split the dataset by taking every fourth chip in a top-down, row-by-row pattern for the test set obtaining a 75/25 split. The size of the chip is a balance between training size and preserving contextual neighbourhood information. The larger the image the more computationally expensive the resulting pipeline becomes however more information about neighbouring features can be encoded. Considering CNN filters, larger chips result in fewer edge cases.

2.4. CNN feature extraction

To learn the texture representation we utilize a CNN where the learning function considers the C-Means classification (Bezdek, Ehrlich, and Full 1984) for an objective discrimination of features. The CNN of choice was a U-Net, with five layers each for the encoder and decoder. Starting with 64 kernels in the encoder, these are doubled for each consecutive layer to 1024 in the final layer and opposite for the decoder. The target of the model is the fuzzy C-Means predictions. We used 2, 8 and 18 clusters to explore the performance of the U-Net. As each fuzzy classification is non-exclusive, a multi-label loss function is used. The loss function combined Dice loss and binary cross entropy Seale et al. (2022).

$$d(i) = 2 \times \frac{\sum(X_i.Y_i)}{\sum(X_i) + \sum(Y_i)}$$

$$D = \frac{1}{C}\sum_{i=1}^{C} d(i)$$

As each class has its respective probability we can calculate the loss for each class, $d(i)$ and average, D. Here $d(i)$ is the dice loss for one class i, X_i is the true probability and Y_i is the predicted. Finally C is the number of classes. Similarly, we calculate binary cross-entropy loss concerning each class and average. The final loss combining the two is a variation of combo loss (Taghanaki et al. 2019).

The final convolutional layer of the CNN produces the same image shape as the input with 64 convolutional activation maps in contrast to the 12 spectral channels. To combat over-fitting we applied random rotation and noise to the training set to increase the variation artificially. In addition, we applied early stopping when the loss was no longer reducing after 15 epochs. There is an increase in loss as the number of fuzzy C-means classes increases as we discriminate between even finer spectral detail.

2.5. Graph construction

Nodes for the graphs are determined by the Simple Linear Iterative Clustering (SLIC) algorithm (Achanta et al. 2012) where each segment is considered a new node. SLIC produces local homogeneous parcels which for encoding geographical neighbours is ideal. We set the target number of segmentations for SLIC as $N = 500$. This choice was made to over-segment the image, resulting in a roughly $1310 m^2$ coverage for each segment, which provides finer granularity while balancing computational cost. Although SLIC dynamically adjusts N based on each sample as it splits or merges segments, we selected $N = 500$ as a practical compromise between segment size and efficiency. This decision was supported by sensitivity analysis (Section 3.4), where $N = 500$ consistently achieved the best trade-off between accuracy and performance compared to $N = 200$ and $N = 800$. We used SLIC for super-pixel generation due to its computational efficiency and ability to produce spatially compact, homogeneous segments, which are well-suited for encoding geographical neighbourhoods in graph structures. SLIC is widely used in both computer vision and remote sensing because it offers fine control over segment size and compactness while maintaining low memory overhead, an important consideration for large Sentinel-2 tiles. Although Sentinel-2 imagery contains multiple spectral channels, we applied SLIC to the RGB bands only. This choice reflects a trade-off between segmentation quality and computational cost: RGB channels capture dominant spatial and textural patterns for defining segment boundaries, while the full multi-band spectral information is incorporated later during CNN feature extraction.

The features of each node are $F_i = \{\bar{A}\}$. Where $\bar{A}$ is the mean values of each of the 64 activation maps from the CNN for each segment. Edges for each segment are determined by connecting to the K geographically nearest segments, where $K = 8$. This strategy was chosen to preserve spatial context, which is critical for remote-sensing semantics: for example, distinguishing a water segment surrounded by ocean from one adjacent to urban areas. Using feature similarity-based K-NN would risk linking distant regions with similar textures, losing neighbourhood information that our pipeline relies on for context-aware embeddings. Geographical adjacency also offers computational advantages, as edge connections remain fixed and do not require iterative recomputation during training. The choice of $K = 8$ was supported by sensitivity analysis (Section 3.4), where this value achieved the best balance between accuracy and efficiency. This work differs from previous approaches by Diao et al. (Diao et al. 2022) because we only include information from the CNN and rely on geographical neighbours to construct edges. The resulting representation is a graph $G(V, E)$ with $N = V$ nodes, each with 64 features F. There are $K \times N$ unweighted edge connections between nodes, approximately 8×500 edges.

2.6. Graph neural network

The graph attention network is three layers with attention mechanisms (Brody, Alon, and Yahav 2021). The final layer produces the C dimensional output for each class predicted. The loss function is categorical cross entropy for each predicted class compared to the averaged C-mean of each segment. The new hidden feature vector for each node, $\mathbf{X}'_i$ can be calculated with the following:

$$\mathbf{X}'_i = \|_{k=1}^{K} \sigma \left(\sum_{j \in N(i) \cup i} a_{ij} \mathbf{W}^k \mathbf{x}_j \right)$$

where k is the number of attention heads, $\|$ denotes vector concatenation, $\sigma(\cdot)$ is an activation function, N is the Neighbourhood of edges to i, $\mathbf{W}^k$ is a matrix of parameters for the k-th attention head and a_{ij} are attention coefficients defined by the following:

$$a_{ij} = \frac{\exp\left(\mathbf{a}^T \text{LeakyReLU}\left(\left[\mathbf{W}\mathbf{x}_i \| \mathbf{W}\mathbf{x}_j \| \mathbf{e}_{ij}\right]\right)\right)}{\sum_{k \in N(i) \cup i} \exp(\mathbf{a}^T \text{LeakyReLU}([\mathbf{W}\mathbf{x}_i \| \mathbf{W}\mathbf{h}_k \| \mathbf{e}_{ik}]))}$$

where $\mathbf{e}_{ik}$ is an edge between node i and connected node k.

Each hidden feature vector for a node has 64 variables, given that our ultimate goal is to visualize and compare similarities we need to maintain or reduce the size of the feature vector in favour of improved computational performance over accuracy. This is achieved by reducing the output feature dimensions of

the GAT layers. In our experiments, the extracted features are the output of the second to last GAT layer, within this layer we constrict the output of the features to eight hidden feature vectors with eight attention heads. Therefore our final representation is a 64-dimensional vector. The third layer reshapes this vector to enable comparison with the targets in the loss function.

Similarly we tested a 3-layer graph convolutional network (GCN) with an embedding vector of 60 hidden features. The choice of testing both networks is to see if the inclusion of an attention mechanism impacts the information in each node in relation to its neighbours.

For these and the above parameters we systematically explored various parameter configurations (e.g. number of GNN edges, U-Net and GCN layers, nearest neighbourhood for SLIC) and selected those that provided the best trade-off between accuracy and computational efficiency, focusing on parameters that had the most significant impact on performance (see Graph layer ablation).

2.7. Graph matching

For each chip in our pipeline, we obtain a graph $G(V, E)$ where each node V is linked by edge E. A node refers to a singular segmentation within the chip (see the yellow outlined segments in Figure 1). Edges are connections to the nearest neighbouring nodes in feature space, X from the second layer of the GNN. The second layer is chosen as it provides contextual information for each segment as it encodes neighbours from two hops away, neighbours of neighbours (this is evaluated in Section 2.9.2). Graph matching finds how similar two graphs are based on their node and edge composition. As our approach utilized many segments our resulting nodes are in the hundreds, therefore, utilizing multi-graph solvers which take into account both vertices and edges is infeasible in both memory and computational complexity. We simplify the matching to a bi-partite graph representation using the Hungarian algorithm to match nodes present in both graphs and calculate the overall similarity based on the distance between matched nodes in feature space X. Applying the matching to create a similarity matrix for every chip processed through UMAP (McInnes, Healy, and Melville 2020) creates a two-dimensional embedding for visualization as follows in Section 2.8.

2.8. 2D projections

We provide the user with an interface that can switch between two modes. In the higher-level mode, users interact with 256×256 pixel chips. In the lower-level mode, users interact with the segments. In both cases, the user interacts through a 2D UMAP projection of the manifold – Figure 2 (top) – and sees the relevant chips or segments in the view below – Figure 2 (bottom).

UMAP requires either a set of vectors representing each data point or a precomputed similarity matrix, typically based on pairwise distances or cosine similarities. Low-level segments are represented by a singular 64 or 60-dimensional vector from the graph attention networks and are readily available for UMAP projection. Higher-level chips are represented as graph structures, where each node corresponds to a 64 or 60-dimensional vector. To compare the two chips, we apply Hungarian matching to align their nodes, producing a one-to-one correspondence between vectors. We then compute the cosine similarity for each matched pair and average these values to obtain an overall similarity score between the two chips. Repeating this process across all chip pairs yields a similarity matrix, which serves as input to UMAP for generating the high-level 2D projection.

2.8.1. Front end (user interface)

The front end provides the interactive 2D manifold view (UMAP) at two levels: (i) chip-level points, and (ii) segment-level points for selected images. Users can pan, zoom, and brush rectangular selections. Selected points are reflected in a linked image panel that renders the corresponding chip thumbnails or super-pixel masks. For performance, the user interface is written using DirectX 11 and C++. The real-time interaction (in the accompanying video) was captured whilst using an NVIDIA RTX 3090.

2.8.2. Back end (model and data handling)

All model training and inference are implemented in PyTorch with GPU acceleration via CUDA. The offline pipeline includes U-Net training for feature extraction, GNN training for context-aware embeddings, and
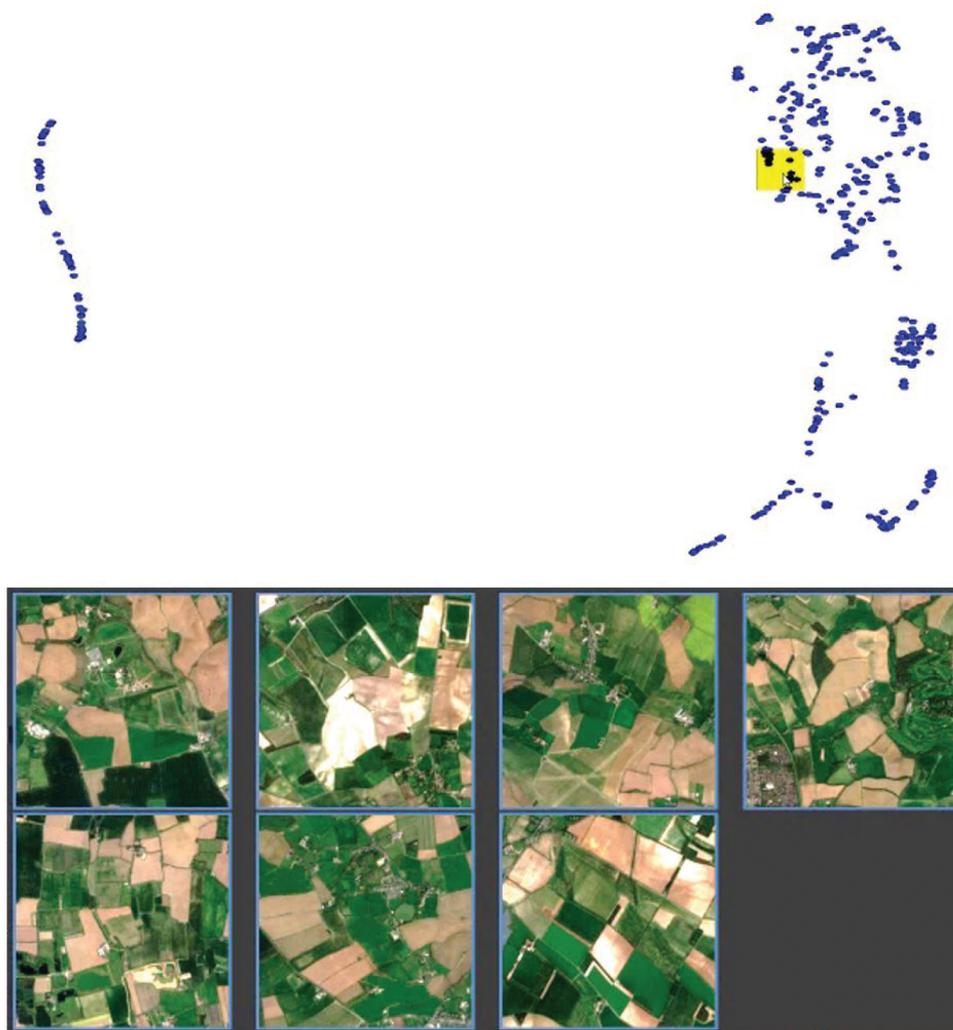
Figure 2. UMAP dimension reduction to 2D on the graph matching output of our entire pipeline. At this level, each point represents one chip. The user interactively highlights a resizeable region which can be dragged across the manifold representation. Chip images represented by the 2D points within the highlight are displayed in the pane below. Training did not use any labelled data. (video frame edited to save space in the paper).

UMAP projection for dimensionality reduction. These activation maps, graph embeddings, and 2D coordinates, are saved to files and accessed by the front-end interface for interactive exploration. During interaction, the tool reads these precomputed files to render chip thumbnails, segmentation masks, and manifold positions in real time. This design separates heavy batch computation (performed offline on an NVIDIA RTX 3090 GPU) from lightweight rendering and selection operations.

2.9. Evaluation

To test the GNN part of our architecture, we define two levels of similarity evaluation between the SLIC super-pixels (segments) – feature-based and context-aware.

2.9.1. Graph evaluation I: feature-based
Under feature-based evaluation the SLIC segments are fed into each GNN, producing feature space, X. For each segment, we compare it to its closest segment within the feature space. The comparison was made using four common similarity measures; grey level occurance matrix (GLCM), linear binary patterns (LBP), Structured similarity index measure (SSIM) and spectral angle mapper (SAM). The first two measures were

calculated via a bounding box that enclosed the segment, the latter two purely on the content within the segment. This test allows us to find the conditions and parameters that produce the best GNN.

2.9.2. Graph evaluation II: context-aware

Under context-aware evaluation, we enhance the evaluation by considering spatial context. For each segment x, we evaluate its similarity not only to y in feature space but also by taking into account the eight nearest spatial neighbours of both segments. The similarity between x and y is then adjusted to ensure that the spatial arrangement of their neighbours is also preserved. We use the Hungarian matching algorithm to find the optimal one-to-one correspondence between the neighbours of x and those of y, to test neighbourhood preservation. This context-aware evaluation leads to a more accurate representation of the similarity between super-pixels, reflecting both feature similarity and spatial proximity. The one-to-one correspondence resulting from the bi-partite matching, leads to nine node pairs, each pair consisting of two segments, either x or a neighbour of x and y or a neighbour of y. Each pair's similarity is then computed with GLCM, LBP, SSIM and SAM and averaged for all nine pairs.

2.9.3. U-Net evaluation

Part of our architecture is a U-Net that provides unsupervised features to feed into the GNN and graph matching. For this evaluation we compare our U-Net architecture to the state-of-the-art approaches of using ResNet-50 and GoogLeNet (Helber et al. 2019). The benchmarked approaches (Helber et al. 2019) were pre-trained on millions of images and then fine-tuned on EuroSat data. We utilize our pre-trained U-Net with two additional linear layers appended for classification. Our U-Net being pre-trained on only 1411 images, from a singular geographical region. If this test achieves an accuracy close to state-of-the-art we can attest that our U-Net, with some fine-tuning, has learnt rich features for the remaining parts of our pipeline. Successful application of the U-Net on a larger and more geographical diverse dataset also would demonstrate geo-generalizable weights.

2.9.4. Tool evaluation

Additional to the above approaches, we also provide an evaluation of our interactive application that allows the user to explore and label the final features produced in two dimensions via UMAP for exploration and labelling. Please refer to the video submitted as a supplementary file for an example exploration of the feature space to label at both the top chip level and the segmentation level. A public version is available at YouTube https://youtu.be/GZl1ebZJgEAYouTube., and is archived at https://doi.org/10.5281/zenodo.16676591. Zenodo (Patel and Jones 2025).

3. Results

3.1. Remote sensing labelling application (see 2.9.4 tool evaluation)

Our pipeline enables advanced interaction within our labelling tool. Referring to the provided video Patel and Jones (2025) of the tool utilized on the dataset, this section presents frames extracts from the video to demonstrate and discuss functionality.

The main contributions are (Section 3.1.1) the tool allows exploration of feature space, X, via a two-dimensional interface where our pipeline has resulted in successful organization of the high-dimensional manifold. The resultant clusters are compact and contain visually related images. (Section 3.1.2) embedded images with large texture gradients have no adverse impacts on the similarity measure using our approach. (Section 3.1.3) we significantly extend previous labelling approaches by enabling this interaction on pixel-based segmentations within the larger images, thus allowing fast labelling at a finer granularity.

3.1.1. Cluster exploration

In Figure 2 (top), the user interacts with the 2D manifold representation of the 256×256 chip-level data by brushing selected points. The brushed points, highlighted in yellow, correspond to specific source images, which are simultaneously displayed in the image pane (Figure 2, bottom). This particular cluster reveals a patchwork of fields. The accompanying video illustrates the manifold's spatial evolution by showing image

chips transitioning through farmland, industrial zones, residential areas, increasing water bodies (e.g. lakes and sea), and other geographical features.

Figure 3 shows more example content of the embedding space. Samples drawn and shown from the clusters contain mainly sea (A), land with water (B), dense built-up areas (C) and farmland (D). Area A in the embedding is consistently majority water content, there is very little presence of land. However, the features within that cluster do not differentiate between any features such as boats or offshore installations. Sample space B includes nearly all water content features present near land, the cluster sits directly between A and D which are exclusively just land or water. This evolution of features shows at the chip level that water content is a highly relevant feature for similarity.

3.1.2. Rotational invariance (Figure 4)

Within our application, the utility of representing our images as segmentations with mean feature aggregation has a two-fold effect. Firstly, rotational invariance is introduced by Hungarian matching disregarding the geographical layout of features, as only segments are compared. Secondly, taking the mean of each feature map removes any encoding of strong gradient changes within the activation maps. For an example of both, we refer to Figure 4. Work produced by (Patel, Jones, and Redfern 2023) shows an example of a large textural and spectral disparity between land and sea. The clustering has been directly influenced by the orientation of this boundary, with the inclusion of outliers containing cloud and water where they follow the same gradient. In comparison our work has ignored strong directional gradients within the image data, e.g. see Figure 3. In Figure 4, we visually selected a cluster also of approximately 50/50 land/water coverage, which indicates there is no obvious common directional gradient.

3.1.3. Segmentation analysis (Figure 5)

A major contribution is enabling finer sub-image labelling based on the segmentations. The user starts by selecting images they wish to examine in detail. A new interactive 2D projection of the feature space, X, is displayed where each point represents a single (SLIC) segmentation within the images. Similar to the image exploration, our approach produces a successful organization of the manifold where segments are very closely related. In Figure 5 we see at the top the 2D interactive display where each point represents one segment within the image data. By brushing multiple
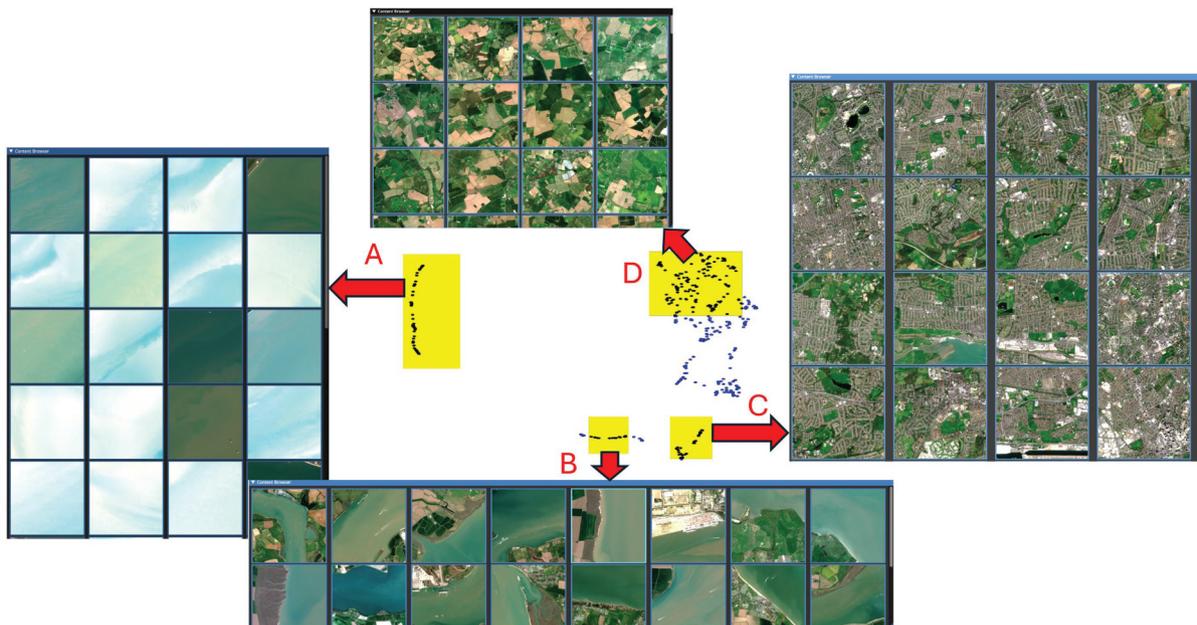

Figure 3. Example of the U-map embedding dimensionality reduction of high dimensional feature space, X, output from the final graph matching stage of our entire pipeline. Training was unsupervised. The images selected by the user highlights are shown in the display pane with the labels A-D explained in Section 3.1.1.
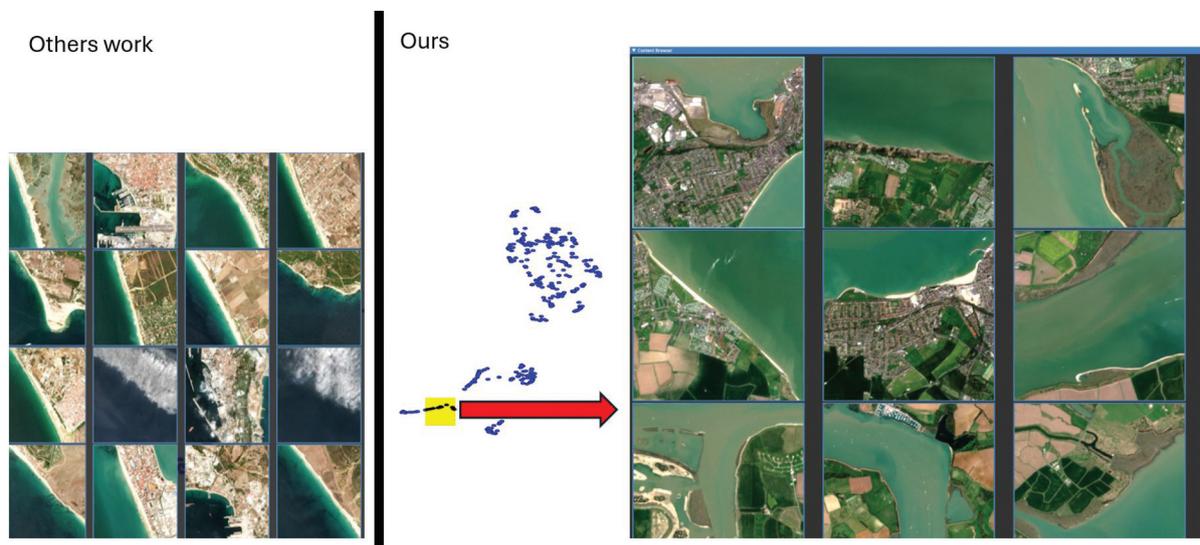
Figure 4. Example of the UMAP embedding space from (Patel, Jones, and Redfern 2023), showing results influenced by strong texture similarities due to edge alignment, and a comparison with our approach, which introduces rotational invariance to remove this strong alignment.

points in 2D, the user will see all corresponding segmentations in the reference images below. This manifold evolves from vegetation on the left to largely urban areas on the right. Here we combine two frames from the video where the left selection displays the vegetation with some urban features (top row of images), and the right selection displays the almost pure urban features (bottom row of images). This empowers the expert to quickly label at a fine level of segmentation (as demonstrated in the video).

The third row shows a further example of extracting features that share common geographical neighbourhoods, in this particular example golf courses. Although unsupervised, the embedding space has clustered larger and smaller golf courses irrespective of their sizes. The video demonstrates an example of how to build up a labelling of the golf course data through selection and filtering data points in the manifold.

The projection at segment level has also clustered the various water bodies within the images (Figure 6). Via exploring the manifold, the user is able to make a selection that segments water features effectively. We found that various different forms and sizes of water bodies including large bodies of water are easily selected via the interface and extracted simultaneously.

3.2. U-Net test (proposed in U-Net evaluation 2.9.3)

The aim of this test is to evaluate whether the U-Net part of our architecture achieves close to state-of-the-art performance to feed into the rest of our graph neural networks, graph matching and interactive application pipeline.

In order to validate the utility and generalizable application of the methodology we want to demonstrate that the trained U-Net (trained using unsupervised C-means clustering) is generalizable. In order to evaluate the U-Net we compare to a benchmark experiment on the EuroSat dataset (Helber et al. 2019). The dataset consists of 27,000 labelled images from Sentinel-2 mission covering ten separate classes. We resized the images from 64×64 to 256×256 to match the training data. We did not apply bottom-of-atmosphere correction. The Eurosat models were pre-trained on 1.2 million images from the ILSVRC-2012 dataset.

We append and train two additional linear layers for class prediction on the EuroSAT dataset. Our aim is to evaluate if the learnt weights are generalizable and whether the model can find an appropriate minimum given a starting feature space. Despite not using a traditional classification model (we use a U-Net) and pre-training on a much smaller dataset, our approach achieves performance comparable to state-of-the-art methods (Table 1). This validates the U-Net component of our architecture.
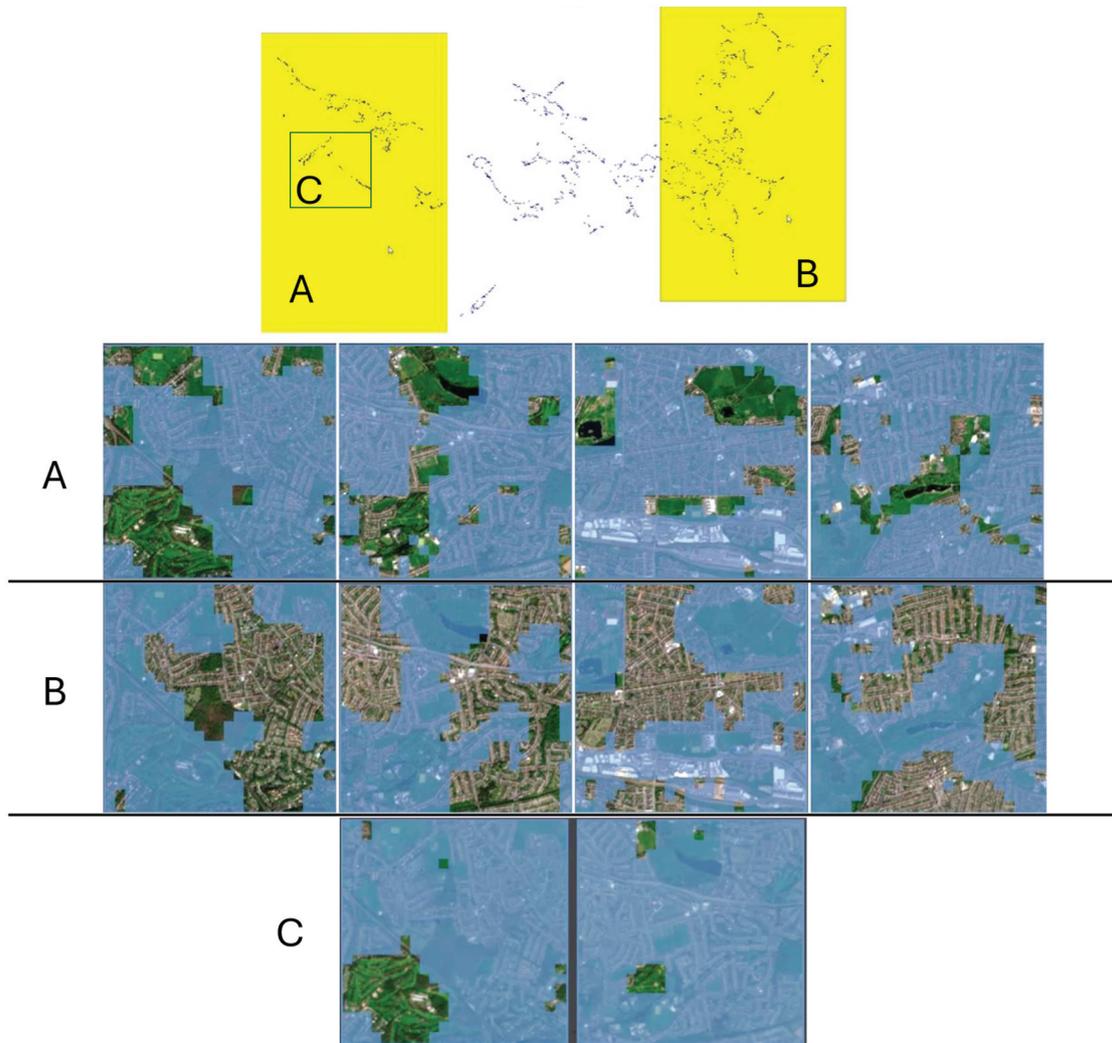
Figure 5. Example (from video) of projecting four images and exploring their segmentations to label as urban or vegetation. The left selection, (A), shows the largely vegetation segments in the first row of images and (B) largely urban development in the same four images, but demonstrated in the second row. (C) is a selection in the manifold of segmentations which related to golf courses and are present on different images as displayed on the third row (the golf course example is in the video).

It is important to note that Table 1 is not intended as a primary result of this work but as a validation step to demonstrate that our U-Net, trained in an unsupervised manner, learns generalizable features suitable for integration into the full pipeline.

3.3. Graph encoding comparison (see 2.9.1 graph evaluation)

We trained GCN and GAT variations of graph networks against C-means clusters with C = 2, 8 and 18. These were evaluated using various similarity measures. GLCM and LBP indicate the textural differences between images. SSIM measures structure, contrast and luminance. SAM is based solely on the mean spectral difference between each segment. This section analyses the construction of our embedding spaces by determining the effect of the different C-means cluster number on the pipeline.

This first experiment computes the measures between each segment, A, and its nearest neighbour, B, in the feature space X (Table 2) and average that over all the segments. This tests whether the GNN places similar segments together in the feature space. The number of C-means clusters exhibits low sensitivity, and we therefore fix C = 8 as the optimal choice for our model.
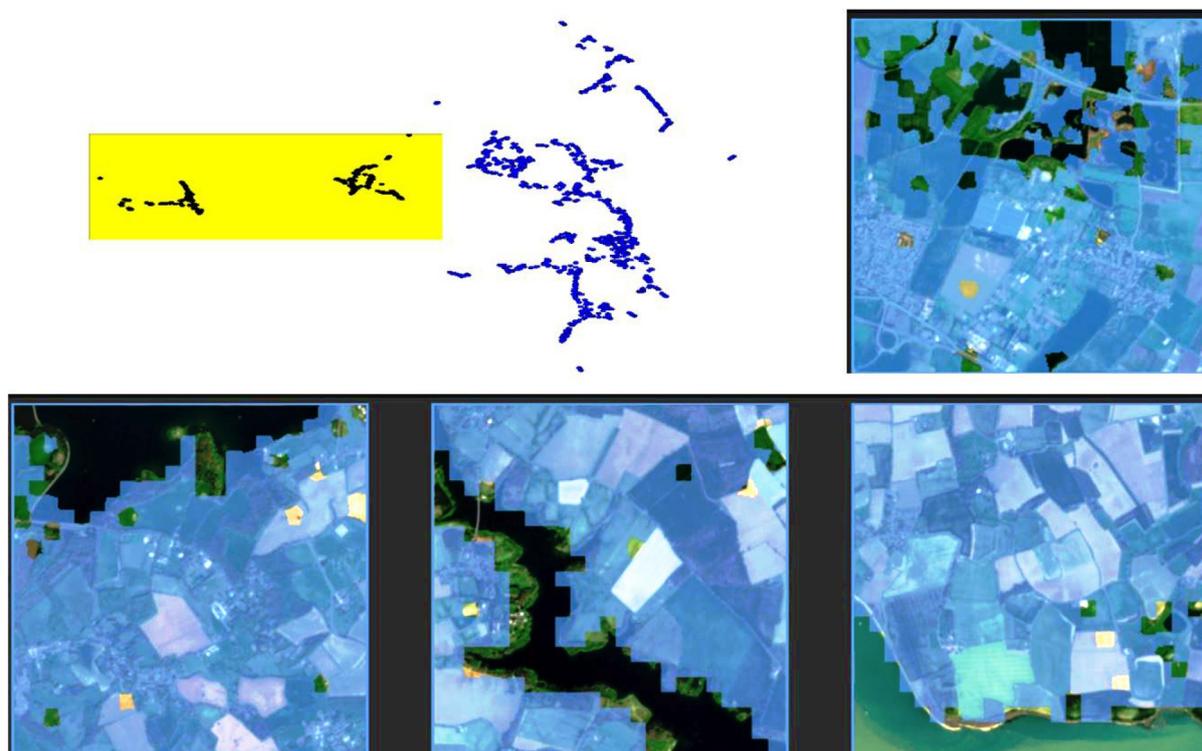
Figure 6. Example of extracting water features at segment level. There are various water features including lakes, rivers and sea. Here, as in Figure 5, the semi-transparent blue indicates the areas masked away, and the full colour segments (which are dark because they are water features) are the selected areas for labelling.

Table 1. Classification accuracies for each model tested, where C is the initial fuzzy C-Means clustering number. EuroSat results are taken from (Helber et al. 2019). These results serve as a validation of the U-Net component rather than the primary contribution, confirming its ability to learn generalizable features for use in our unsupervised labelling pipeline.

Method	Accuracy
EuroSat ResNet-50	98.57
EuroSat GoogleLeNet	98.18
C=2 linear layers	93.24
C=8 linear layers	97.43
C=18 linear layers	93.42

Table 2. Average of each similarity measure over every pair of nearest neighbouring segments in feature space, X, created by each method. This corresponds to Section 2.9.1 feature-based graph evaluation.

Model	GLCM ↓	LBP ↑	SSIM ↑	SAM ↓
GCN 2	16.2770	0.8571	0.9120	0.3321
GCN 8	**15.4685**	**0.8716**	**0.9287**	**0.2887**
GCN 18	15.5322	0.8671	0.9243	0.3017
GAT 2	16.0025	0.8633	0.9201	0.3112
GAT 8	15.5676	0.8704	0.9256	0.2942
GAT 18	15.7663	0.8673	0.9237	0.2992

Table 3. Average of each similarity measure over every pair of nearest neighbouring segments. The similarity between each segment takes into account the context of its local geographical neighbourhood as described in the referring text. This corresponds to Section 2.9.2 context-aware graph evaluation.

Model	GLCM ↓	LBP ↑	SSIM ↑	SAM ↓
GCN 2	14.4610	0.8983	0.9589	0.2152
GCN 8	**13.7904**	**0.9034**	**0.9624**	**0.2064**
GCN 18	13.7278	0.9033	0.9620	0.2066
GAT 2	14.2259	0.9000	0.9560	0.2122
GAT 8	14.0145	0.9018	0.9613	0.2092
GAT 18	14.0557	0.9013	0.9611	0.2100

The second experiment will find the eight spatially closest segmented neighbours in the original image to A. It also finds the eight spatially closest segment neighbours to B. Hungarian matching is then used to pair segments between these two neighbourhoods, forming nine matched pairs. The similarity measures are computed for each pair to assess how similar the neighbourhoods are. This experiment tests whether segments with similar local contexts, implicitly captured by the two graph layers, are also close in the learned feature space. Ideally, the feature space should position segments near others that share similar neighbourhood structures. For example, a small body of water surrounded by urban segments should be placed near other similarly situated water bodies.

Overall the aim of this test is to evaluate that our pipeline is able to place segments of a certain type and surrounding close within the feature space as this lends itself to being the best interaction for a user by enforcing local structural awareness. The best similarity results were for eight clusters and GCN type was better than GAT, see Table 3. From our results presented in the video it is possible to see that segments are contextually close making the labelling easier. The graph convolutional network, trained on eight C-means clusters seems to be the best balance in our tests and was therefore used in our pipeline, although again there is low sensitivity to cluster number. Overall these models take into account the neighbourhood and texture more than the individual segments or spectral differences.

3.4. Parameter sensitivity

In this section we explore the parameter sensitivity for two key values within the pipeline. These two parameters are the initial K value and N for SLIC during the graph construction phase, Section 2.5. The variations in parameters are tested via our context-aware approach to segmentation comparison detailed in Section 2.9.2. We also report results when utilizing only the CNN output and the first layer of the GCN. These tests are compared against the best pipeline (GCN 8) where $K = 8$ and $N = 500$ segments in the SLIC algorithm.

We used eight edges during the graph generation phase. Here we compare results for $K \in \{4, 8, 12\}$. Our pipeline shows an improvement in for larger number of edges (K) in the neighbourhood (Table 4). Increasing the number of edges (K) increases the GCN training time and graph matching. For example, given the Hungarian algorithm's complexity of $O(N^3)$, where N is the sum of all nodes in both graphs, the complexity for $K = 8$ is 16^3 whilst $K = 12$ is 24^3. Correspondingly, for the GCN training times for $K = 8$ was 1.6 hours and $K = 12$ was 6.6 hours. In our tests, larger K produces more accuracy, but at significantly slower training times, and therefore we used $K = 8$ as a good balance.

Here we report values of $N \in 200, 500, 800$ for the number of SLIC segments. Our initial choice, $N = 500$, was to cover an approximate area of $1300m^2$ for ease of interpretability and efficiency when visualized for

Table 4. Comparison of varying K, during the graph construction phase (Section 2.5).

K	GLCM ↓	LBP ↑	SSIM ↑	SAM ↓
4	14.2846	0.8992	0.9583	0.2140
8	13.7904	0.9034	0.9624	0.2064
12	**13.2240**	**0.9075**	**0.9649**	**0.1988**

the user (the mean average size of segments is $256 \times 256 \div 500 = 131$ — approximately 11 by 11 pixels and each pixel is $10m^2$ in Sentinel-2). A smaller area allows finer labelling at the cost of speed, namely projecting and labelling more points. This choice for N was also optimal when considering local neighbourhood encoding using the different measures, see Table 5. For smaller N values, we see that comparing segments via GLCM shows better results however, over all measures $N = 500$ is the most accurate.

3.5. Graph layer ablation

Finally, we validate our architecture by conducting an ablation study on the use of embeddings from our graph output. Our architecture produces graph outputs at three stages: (1) during graph generation, which combines the CNN U-Net output and SLIC; (2) at the output of GCN layer 1; and (3) at the output of GCN layer 2, which is the version reported throughout this work. The output of GCN layer 3 is a learnt reshaping from the second layer, from hidden dimension to C-means predictions, used only for the loss function. As such, it is not suitable for use as an embedding.

Table 6 demonstrates that selecting GCN layer 2 as the embedding layer yields optimal performance on three of the metrics related to texture and spectral similarity, which are particularly relevant to our interactive pipeline. The only exception is SSIM, where the combination of CNN U-Net output and SLIC performs best.

3.6. Computational complexity and runtime efficiency

The proposed pipeline was designed with efficiency in mind, balancing accuracy and computational cost through parameter tuning (Section 3.4). The complexity of each stage is as follows: SLIC segmentation operates in approximately $O(P)$ time, where P is the number of pixels, and is highly efficient for large Sentinel-2 tiles. Fuzzy C-Means clustering introduces iterative updates but remains tractable for moderate cluster counts ($C \in \{2, 8, 18\}$). Using THOP, our U-Net requires 55 GFLOPs per forward pass for a 256×256 input with 12 Sentinel-2 channels. The model contains 31 M trainable parameters. Graph construction scales with $O(N + K \cdot N)$, where N is the number of super-pixels and K the number of edges per node (here $K = 8$). Message passing in the GNN is linear in the number of edges, $O(|E|)$, and was implemented with efficient batching. The GNN contains 17K parameters and requires 415 MFLOPs per forward pass. Hungarian matching introduces the highest theoretical complexity at $O(n^3)$ for n nodes per graph (approximately $n \approx 500$), but this step is applied only during similarity computation and is GPU-accelerated. Dimensionality reduction via UMAP scales well for interactive use and was precomputed for large batches.

On our test hardware (NVIDIA RTX 3090 GPU, 24 GB RAM), training the U-Net required approximately 10 hours for 600 epochs, and the GNN training completed in 1.5 hours for 300 epochs. Inference for a single tile (42×42 chips comprising 256×256 pixels) through the pipeline (segmentation, feature extraction, graph generation, graph inference, Hungarian matching, and embedding) requires approximately 23 minutes. This time is dominated by the $O(n^3)$ Hungarian matching process (194,481 pairs of chips) which takes 20 minutes (Table 7),

Table 5. Comparison of varying SLIC parameter N, used during the graph construction phase Section 2.5.

SLIC	GLCM ↓	LBP ↑	SSIM ↑	SAM ↓
200	**12.4336**	0.8888	0.9166	0.3608
500	13.7904	**0.9034**	**0.9624**	**0.2064**
800	14.8003	0.8604	0.9404	0.2819

Table 6. Comparison of utilizing different outputs for the final graph representation, the graph construction phase (Section 2.5) and the first layer of the GCN. These are compared to the ideal pipeline, GCN with 8 C-means.

Image Representation	GLCM ↓	LBP ↑	SSIM ↑	SAM ↓
Graph Generation	14.6613	0.8970	**0.9853**	0.2165
Layer 1	14.3951	0.8988	0.9590	0.2147
GCN Layer 2	**13.7904**	**0.9034**	0.9624	**0.2064**

Table 7. Example runtime breakdown of the full processing pipeline on 1764 chips (each 256 × 256 pixels).

Step	Time
Segmentation	2 min 20 s
Feature extraction (U-Net)	6 s
Graph generation (parallelized)	8 s
GNN	6 s
Hungarian matching, (194,481 pairs, parallelized)	20 min
Embedding (UMAP)	15 s

however is only computed once for a dataset. The application interaction achieves a frame rate update of 60 frames per second, although this is variably dependent on the amount of points selected by the user via the brushing interaction.

4. Discussion

Our pipeline creates a 2D interactive interface that enables users to apply labels to segments and chips extracted from remote sensing (RS) images. We demonstrate that unlabelled data can be effectively leveraged using C-means clustering, SLIC segmentation, and a U-Net architecture, all of which contribute to the loss function of a graph convolutional network (GCN). The GCN takes as input the SLIC segments and the final layer of the U-Net, allowing it to learn representations that capture both individual segment features and their spatial context. The resulting graph embeddings are compared using Hungarian matching to assess neighbourhood similarity. These comparisons are used to construct a similarity matrix, which is then projected into 2D using UMAP to power the interactive interface.

By exploring hyperparameters, we arrive at an optimal model and architecture that, as shown in the accompanying video, enables intuitive and efficient interaction with both high-level chips and low-level segments. We validated key components of the model and provided a quantitative assessment of its performance including sensitivity and ablation study.

The high ingestion rate of remote sensing (RS) data and the need for labelled datasets remain major challenges – especially in emerging or niche application areas where accurate, up-to-date labels are scarce. Creating large, high-quality datasets typically requires costly expert input. To address this, our proposed application framework reduces labelling effort by enabling users to find similar features across RS images. Presenting data as image chips, rather than traditional map views, streamlines the labelling of large areas. While map-based projections offer geographic context, chips allow users to more easily assess visual and contextual similarities. Future work could integrate both views through a coordinated interface.

Remote sensing (RS) data captures the physical properties of surface materials, which reflect light at varying intensities depending on the recorded wavelength. Within a single $10m^2$ pixel, multiple materials may contribute to the signal, resulting in mixed spectral responses, therefore, spectral unmixing was the original inspiration behind our generalizable CNN model. Our use of fuzzy clustering emulates this principle by allowing soft associations between segments and composite material types. However, an open question remains: could the pipeline be further improved by integrating spectral unmixing algorithms (Feng et al. 2022) commonly used in the RS community? Alternatively, incorporating a priori knowledge of material reflectance properties could guide centroid selection to better target or generalize the labelling process.

The graph matching process within this pipeline and the tests poses an interesting computational problem. Whilst Hungarian matching can be processed on graphics cards reducing time taken it still remains a costly problem to solve. The complexity is $O(n^3)$ where n is the count of nodes in both graphs. In order to save time this process could be estimated via an AI model. Similarly UMAP and t-SNE could be solved by training parametric models (Sainburg, McInnes, and Gentner 2021; Svantesson et al. 2023). Large dataset projections could be calculated offline to user interaction however branched projections of many segments could benefit greatly from parametric models.

At the low-level we allow users to label segments. Even lower pixel-level classification would require future work to improve the segmentations where only extremely similar or identical pixels are grouped. This could impact the time complexity of the graph matching as more segmentations would be required also

placing more demand on dimensionality reduction. Alternatively segments could be organized hierarchically and presented to a user for accurate pixel level classification, therefore only computing relevant information when needed.

Disclosure statement

Funding

Data statement

Data supporting this study are included within the article and its supporting materials. The work also utilized two publicly available datasets: the Eurosat dataset by Helber et al. ([Helber et al. 2019]) and the Sentinel-2 Water Edges Dataset (SWED) (Seale et al. 2022). The Eurosat data is available publicly here (https://github.com/phelber/eurosat). The SWED dataset is available from https://openmldata.ukho.gov.uk/.

References

Achanta, R., A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. 2012. "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 340 (11): 0 2274–2282. https://doi.org/10.1109/TPAMI.2012.120

Ahmadi, H., and E. Pekkan. 2021. "Fault-Based Geological Lineaments Extraction Using Remote Sensing and GIS—A Review." *Geosciences* 110 (5): 0 183. https://doi.org/10.3390/geosciences11050183

Ali, M., R. Borgo, and M. W. Jones. 2021. "Concurrent Time-Series Selections Using Deep Learning and Dimension Reduction." *Knowledge-Based Systems* 233:0 107507, 12. https://doi.org/10.1016/j.knosys.2021.107507

Ali, M., M. W. Jones, X. Xie, and M. Williams. 2019. "Timecluster: Dimension Reduction Applied to Temporal Data for Visual Analytics." *The Visual Computer* 350 (6): 0 1013–1026. 6. https://doi.org/10.1007/s00371-019-01673-y

Bezdek, J. C., R. Ehrlich, and W. Full. 1984. "FCM: The Fuzzy C-Means Clustering Algorithm." *Computers & Geosciences* 100 (2–3): 0 191–203. https://doi.org/10.1016/0098-3004(84)90020-7

Brody, S., U. Alon, and E. Yahav. 2021. How Attentive Are Graph Attention Networks? *arXiv preprint arXiv:2105.14491*, https://doi.org/10.48550/arXiv.2105.14491

Cao, Y., C. Wang, Z. Li, L. Zhang, and L. Zhang. 2010. "Spatial-Bag-Of-Features." 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 3352–3359. IEEE. https://doi.org/10.1109/CVPR.2010.5540021

Chaudhuri, B., B. Demir, S. Chaudhuri, and L. Bruzzone. 2018. "Multilabel Remote Sensing Image Retrieval Using a Semisupervised Graph-Theoretic Method." *IEEE Transactions on Geoscience and Remote Sensing* 560 (2): 0 1144–1158. https://doi.org/10.1109/TGRS.2017.2760909

Chaudhuri, U., B. Banerjee, A. Bhattacharya, and M. Datcu. 2022. "Attention-Driven Graph Convolution Network for Remote Sensing Image Retrieval." *IEEE Geoscience and Remote Sensing Letters* 19:0 1–5. https://doi.org/10.1109/LGRS.2021.3105448

Chen, S., S. Zhong, B. Xue, X. Li, L. Zhao, and C.-I. Chang. 2020. "Iterative Scale-Invariant Feature Transform for Remote Sensing Image Registration." *IEEE Transactions on Geoscience and Remote Sensing* 590 (4): 0 3244–3265. https://doi.org/10.1109/TGRS.2020.3008609

Diao, Q., Y. Dai, C. Zhang, Y. Wu, X. Feng, and F. Pan. 2022. "Superpixel-Based Attention Graph Neural Network for Semantic Segmentation in Aerial Images." *Remote Sensing* 140 (2): 305. https://doi.org/10.3390/rs14020305

Draganov, A., and S. Dohn. 2023. Relating tSNE and UMAP to Classical Dimensionality Reduction. *arXiv preprint arXiv:2306.11898*, https://doi.org/10.48550/arXiv.2306.11898

Feng, X.-R., H.-C. Li, R. Wang, Q. Du, X. Jia, and A. Plaza. 2022. "Hyperspectral Unmixing Based on Nonnegative Matrix Factorization: A Comprehensive Review." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15:0 4414–4436. https://doi.org/10.1109/JSTARS.2022.3175257

Halladin-Dabrowska, A., A. Kania, and D. Kopeć. 2019. "The T-SNE Algorithm as a Tool to Improve the Quality of Reference Data Used in Accurate Mapping of Heterogeneous Non-Forest Vegetation." *Remote Sensing* 120 (1): 0 39. https://doi.org/10.3390/rs12010039

Helber, P., B. Bischke, A. Dengel, and D. Borth. 2019. "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 120 (7): 2217–2226. https://doi.org/10.1109/JSTARS.2019.2918242

Hou, D., Z. Miao, H. Xing, and H. Wu. 2019. "V-RSIR: An Open Access Web-Based Image Annotation Tool for Remote Sensing Image Retrieval." *IEEE Access* 7:0 83852–83862. https://doi.org/10.1109/ACCESS.2019.2924933

Huang, L., C. Chen, W. Li, and Q. Du. 2016. "Remote Sensing Image Scene Classification Using Multi-Scale Completed Local Binary Patterns and Fisher Vectors." *Remote Sensing* 80 (6): 0 483. https://doi.org/10.3390/rs8060483

Huang, X., X. Liu, and L. Zhang. 2014. "A Multichannel Gray Level Co-Occurrence Matrix for Multi/Hyperspectral Image Texture Representation." *Remote Sensing* 60 (9): 0 8424–8445. https://doi.org/10.3390/rs6098424

Imbriaco, R., C. Sebastian, E. Bondarev, and P. H. N. de with. 2019. "Aggregated Deep Local Features for Remote Sensing Image Retrieval." *Remote Sensing* 110 (5): 493. https://doi.org/10.3390/rs11050493

Jégou, H., M. Douze, C. Schmid, and P. Pérez. 2010. "Aggregating Local Descriptors into a Compact Image Representation." 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 3304–3311. IEEE. https://doi.org/10.1109/CVPR.2010.5540039

Kang, J., R. Fernandez-Beltran, D. Hong, J. Chanussot, and A. Plaza. 2021. "Graph Relation Network: Modeling Relations Between Scenes for Multilabel Remote-Sensing Image Classification and Retrieval." *IEEE Transactions on Geoscience and Remote Sensing* 590 (5): 0 4355–4369. https://doi.org/10.1109/TGRS.2020.3016020

Karthikeyan, L., I. Chawla, and A. K. Mishra. 2020. "A Review of Remote Sensing Applications in Agriculture for Food Security: Crop Growth and Yield, Irrigation, and Crop Losses." *Journal of Hydrology* 586:0 124905. https://doi.org/10.1016/j.jhydrol.2020.124905

Li, J., L. Meng, B. Yang, C. Tao, L. Li, and W. Zhang. 2021. "LabelRS: An Automated Toolbox to Make Deep Learning Samples from Remote Sensing Images." *Remote Sensing* 130 (11): 0 2064. https://doi.org/10.3390/rs13112064

Li, Y., J. Ma, and Y. Zhang. 2021. "Image Retrieval from Remote Sensing Big Data: A Survey." *Information Fusion* 67:0 94–115. https://doi.org/10.1016/j.inffus.2020.10.008

McInnes, L., J. Healy, and J. Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426*, 2020. https://doi.org/10.48550/arXiv.1802.03426

Napoletano, P. 2018. "Visual Descriptors for Content-Based Retrieval of Remote-Sensing Images." *International Journal of Remote Sensing* 390 (5): 0 1343–1376. https://doi.org/10.1080/01431161.2017.1399472

Patel, T., and M. W. Jones. 2025. "Video Supplement to: 'Leveraging Convolutional and Graph Networks for an Unsupervised Remote Sensing Labelling Tool'." https://doi.org/10.5281/zenodo.16676591 Video supplement.

Patel, T., M. W. Jones, and T. Redfern. 2023. "Manifold Explorer: Satellite Image Labelling and Clustering Tool with Using Deep Convolutional Autoencoders." *Algorithms* 160 (10): 0 469. https://doi.org/10.3390/a16100469

Qi, X., P. Zhu, Y. Wang, L. Zhang, J. Peng, M. Wu, J. Chen, X. Zhao, N. Zang, and P. T. Mathiopoulos. 2020. "MLRSNet: A Multi-Label High Spatial Resolution Remote Sensing Dataset for Semantic Scene Understanding." *ISPRS Journal of Photogrammetry and Remote Sensing* 169:0 337–350. https://doi.org/10.1016/j.isprsjprs.2020.09.020

Ranchin, T., and L. Wald. 1993. "The Wavelet Transform for the Analysis of Remotely Sensed Images." *International Journal of Remote Sensing* 140 (3): 0 615–619. https://doi.org/10.1080/01431169308904362

Romero, A., C. Gatta, and G. Camps-Valls. 2015. "Unsupervised Deep Feature Extraction for Remote Sensing Image Classification." *IEEE Transactions on Geoscience and Remote Sensing* 540 (3): 0 1349–1362. https://doi.org/10.1109/TGRS.2015.2478379

Sainburg, T., L. McInnes, and T. Q. Gentner. 2021. "Parametric UMAP Embeddings for Representation and Semisupervised Learning." *Neural Computation* 330 (11): 0 2881–2907. https://doi.org/10.1162/neco_a_01434

Seale, C., T. Redfern, P. Chatfield, C. Luo, and K. Dempsey. 2022. "Coastline Detection in Satellite Imagery: A Deep Learning Approach on New Benchmark Data." *Remote Sensing of Environment* 278:0 113044. https://doi.org/10.1016/j.rse.2022.113044

Sivic, J., and A. Zisserman. 2003. "Video Google: A Text Retrieval Approach to Object Matching in Videos." In *IEEE International Conference on Computer Vision*, 1470–1477. Vol. 2. Los Alamitos, CA, USA: IEEE Computer Society. https://doi.org/10.1109/ICCV.2003.1238663

Song, J., S. Gao, Y. Zhu, and C. Ma. 2019. "A Survey of Remote Sensing Image Classification Based on CNNs." *Big Earth Data* 30 (3): 0 232–254. https://doi.org/10.1080/20964471.2019.1657720

Sumbul, G., and B. Demir. 2021. "A Novel Graph-Theoretic Deep Representation Learning Method for Multi-Label Remote Sensing Image Retrieval." 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 266–269. IEEE. https://doi.org/10.1109/IGARSS47720.2021.9554466

Sun, Q., and S. Bourennane. 2020. "Hyperspectral Image Classification with Unsupervised Feature Extraction." *Remote Sensing Letters* 110 (5): 0 475–484. https://doi.org/10.1080/2150704X.2020.1731769

Sun, Y., L. Lei, D. Guan, G. Kuang, Z. Li, and L. Liu. 2025. "Locality Preservation for Unsupervised Multimodal Change Detection in Remote Sensing Imagery." *IEEE Transactions on Neural Networks and Learning Systems* 360 (4): 0 6955–6969. https://doi.org/10.1109/TNNLS.2024.3401696

Sun, Y., L. Lei, Z. Li, G. Kuang, and Q. Yu. 2025. "Detecting Changes Without Comparing Images: Rules Induced Change Detection in Heterogeneous Remote Sensing Images." *ISPRS Journal of Photogrammetry and Remote Sensing* 230:0 241–257. https://doi.org/10.1016/j.isprsjprs.2025.09.009

Svantesson, M., H. Olausson, A. Eklund, and M. Thordstein. 2023. "Get a New Perspective on EEG: Convolutional Neural Network Encoders for Parametric T-SNE." *Brain Sciences* 130 (3): 0 453. https://doi.org/10.3390/brainsci13030453

Taghanaki, S. A., Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, and G. Hamarneh. 2019. "Combo Loss: Handling Input and Output Imbalance in Multi-Organ Segmentation." *Computerized Medical Imaging and Graphics* 75:0 24–33. https://doi.org/10.1016/j.compmedimag.2019.04.005

Tang, X., X. Zhang, F. Liu, and L. Jiao. 2018. "Unsupervised Deep Feature Learning for Remote Sensing Image Retrieval." *Remote Sensing* 100 (8): 1243. https://doi.org/10.3390/rs10081243

Wang, G., B. Fan, S. Xiang, and C. Pan. 2017. "Aggregating Rich Hierarchical Features for Scene Classification in Remote Sensing Imagery." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 100 (9): 0 4104–4115. https://doi.org/10.1109/JSTARS.2017.2705419

Wang, Y., J. Song, K. Zhou, and Y. Liu. 2021. "Unsupervised Deep Hashing with Node Representation for Image Retrieval." *Pattern Recognition* 112:0 107785. https://doi.org/10.1016/j.patcog.2020.107785

Weiss, M., F. Jacob, and G. Duveiller. 2020. "Remote Sensing for Agricultural Applications: A Meta-Review." *Remote Sensing of Environment* 236:0 111402. https://doi.org/10.1016/j.rse.2019.111402

Yan, Y., H. Hao, B. Xu, J. Zhao, and F. Shen. 2020. "Image Clustering via Deep Embedded Dimensionality Reduction and Probability-Based Triplet Loss." *IEEE Transactions on Image Processing* 29:0 5652–5661. https://doi.org/10.1109/TIP.2020.2984360

Yang, Y., and S. Newsam. 2008. "Comparing SIFT Descriptors and Gabor Texture Features for Classification of Remote Sensed Imagery." 2008 15th IEEE international conference on image processing, 1852–1855. IEEE. https://doi.org/10.1109/ICIP.2008.4712139

Yu, H., F. Yao, W. Lu, N. Liu, P. Li, H. You, and X. Sun. 2022. "Text-Image Matching for Cross-Modal Remote Sensing Image Retrieval via Graph Neural Network." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, San Diego, CA, USA, 812–824. https://doi.org/10.1109/JSTARS.2022.3231851

Zhang, L., and L. Zhang. 2022. "Artificial Intelligence for Remote Sensing Data Analysis: A Review of Challenges and Opportunities." *IEEE Geoscience and Remote Sensing Magazine* 100 (2): 0 270–294. https://doi.org/10.1109/MGRS.2022.3145854

Zhang, M., Q. Cheng, F. Luo, and L. Ye. 2021. "A Triplet Nonlocal Neural Network with Dual-Anchor Triplet Loss for High-Resolution Remote Sensing Image Retrieval." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14:0 2711–2723. https://doi.org/10.1109/JSTARS.2021.3058691

Zhang, Y., S. Yan, L. Zhang, and B. Du. 2024. "Fast Projected Fuzzy Clustering with Anchor Guidance for Multimodal Remote Sensing Imagery." *IEEE Transactions on Image Processing* 33:4640–4653. https://doi.org/10.1109/TIP.2024.3444323

Zhu, Q., Y. Zhong, B. Zhao, G.-S. Xia, and L. Zhang. 2016. "Bag-of-Visual-Words Scene Classifier with Local and Global Features for High Spatial Resolution Remote Sensing Imagery." *IEEE Geoscience and Remote Sensing Letters* 130 (6): 0 747–751. https://doi.org/10.1109/LGRS.2015.2513443