Mass labelling Remote Sensing Images: A visualisation and machine learning combinatory approach.

Tulsi Patel

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Doctor of Philosophy

Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

December 15, 2025

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ... [redacted] ..... (candidate)

Date .....15/12/2025.....

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed ..... [redacted] ..... (candidate)

Date .....15/12/2025.....

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... [redacted] ..... (candidate)

Date .....15/12/2025.....

Abstract

Deep learning and computer vision have made significant advancements in the field of the automated analysis of Remote Sensing imagery. However, in order to maximise the utility of satellite derived Remote sensing imagery, numerous challenges must be addressed. Firstly, the complexity of the Earth's surface, in conjunction with its atmosphere, introduces variation within data, requiring expert knowledge to interpret, label and ensure accuracy. Secondly, a large volume of imagery data is generated every day from a single sensing instrument and there are multiple instruments continuously monitoring the earth's surface. Only a very small percentage of data has been accurately labelled, and that which has been is targeted towards a specific domains and of limited temporal resolution. In this thesis, we identify the need for and explore a new remote image labelling framework. We utilise visualisation, unsupervised machine learning and remote sensing methodologies to enable users to create large-scale, labelled remote sensing datasets. Our approach combines three main components - Deep convolutional auto-encoders, manifold learning and an interactive data labelling application. The deep convolutional auto-encoder learns a condensed and informative representations satellite imagery. The manifold learning condenses this representation down into two dimensions, which then supports visualisation techniques that help to convey patterns and representations to a labeller. Graph neural networks are then introduced to further enhance the spatial encoding of geographical features within imagery, and the use of super-pixel representations allow users to create full segmentation labels. Throughout the thesis, we present an evolving visualisation application in which we explore the feature encodings provided by our deep learning pipelines. We introduce a novel methodology for branching and merging datasets, providing a more fine-grained and expressive labelling experience for users. Overall, this thesis presents novel and innovative methods for labelling remote sensing images, using techniques from visualisation, computer vision and the remote sensing domains.

Acknowledgements

I am grateful to many members of the Swansea community, but none more so than my supervisor and advisor, Professor Mark Jones, for both the Master's and Ph.D. candidature. They have supported me without condition through all challenges. This gratitude is also extended to my industrial supervisor, Dr. Thomas Redfern, for his guidance, support, and motivation throughout the thesis, as well as to his predecessor, Dr. Catherine Seale. I would also like to thank Dr Mike Edwards for the advice and support.

I would also like to express my gratitude to my friends in the computational foundry: Mr. Ben-Lloyd Roberts, Mr. Floyd Hepburn, Mr. Luke Thomas, and Dr. Anna Carter, as well as all the other wonderful friends I have made along the way.

Last but not least, I would like to thank my family for their support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Satellite remote sensing is the observation of the earth's surface from space hosted sensors. Advancements in sensor technology and increased satellite missions have seen a large growth in imagery captured and available for analysis. The Earth is diverse and contains features that evolve over time, at a range of geographical scales (from multiple centimetres to kilometer scale). Application of these images is wide spread in industrial, ecological, agricultural, defence and intelligence domains.

Estimates of globally archived data currently exceeding one exabyte [10]. In order to fully capitalise on the vast quantity of data, automatic methods are crucial. This volume of data is impractical for manual interpretation, annotation and analysis. Therefore, in order to fully exploit the information contained within this large (and growing) archive of data, automatic methods for image interpretation, labelling and information extraction are required. In practice however, fully automated image interpretation and labelling is infeasible due to the diversity of the Earth environment, changes in atmospheric and light conditions and domain specific classification requirements. These challenges render even major applications such as road or ship detection unable to be fully automated, datasets for these applications still have varying degrees of accuracy with ambiguous labelling [11, 12]. More accurate and state-of-the-art models such as SatVit require a large labelled dataset to be able to report the performance that they obtain [13]. Common datasets in remote sensing only contain a few thousand images compared to millions used to pre-train SatVit [2, 14, 15]. Even with advanced pre-trained models the varying domains require uniquely labelled data makes, which makes methods such as transfer learning difficult. It is crucial that newer methodologies for labelling remotely sensed images are developed in order to keep up with the ingestion rate, create domain specific

datasets and utilise more recent computer vision models to their fullest.

Manual image labelling remains common in remote sensing for small datasets: expert analysts typically assign predefined classes to pre-selected geographic regions. These classifications may be applied at the scene level (one label for an entire image area), at the pixel/segmentation level (categorical land-cover values for every pixel) or else at the object level (e.g. bounding boxes surrounding a feature of interest e.g. islands). Datasets such as LandCoverNet—where multiple experts labelled the same data and uncertainty scores were recorded when consensus was unattainable—illustrate both the value and the limits of expert labelling which arises from the ambiguity of image interpretation [16].

Semi-automated and automated approaches sit on a spectrum between manual labelling and fully unsupervised methods. Image-retrieval systems speed up labelling by returning visually similar images for rapid human annotation, but they only convey similarity to the query image and do not capture broader relationships within the dataset [17]. Clustering or automated-ML approaches can generate pixel-level groupings that experts then refine [3], while active-learning, real-time detectors and interactive segmentation tools (for example, Segment Anything and one-shot/few-shot detectors like YOLO variants) reduce human effort by prioritising uncertain or informative samples for annotation [18–20]. These tools are powerful but have practical limitations: pretrained models and few-class detectors introduce label-space rigidity (models only predict the classes they were trained on), crowd-sourced labels often lack reliability and fine granularity, and one-shot/few-shot detectors can miss rare or unique samples that are important for downstream tasks [21, 22].

Label-creation toolchains for remote sensing increasingly include preprocessing steps (cloud masking, water masks, corrupt-data removal) and export options that ease dataset production [7]. However, many labelling systems provide only a sparse set of bounding boxes or coarse labels relative to the true feature richness of images [23]. Pre-training and transfer approaches scale well but reduce flexibility: they embed assumptions about class definitions that may not hold in a new domain [24]. Similarly, active learning loops are sensitive to annotation noise——expert disagreement or low-quality crowd labels can degrade model performance and selection strategies [25].

Unsupervised methods avoid dependence on existing labels and the continual annotation-revision cycle [26]. Unlike supervised approaches, which require extensive human labelled datasets and frequent updates as classification requirements evolve, unsupervised models learn the underlying patterns and relationships directly from the imagery itself. The independence

from labelled data makes unsupervised techniques desirable for remote sensing data, where ground truth is often sparse, expensive or inconsistent [27]. Furthermore unsupervised techniques can reveal structure of classes that predefined taxonomies have yet to incorporate [28]. As a result, unsupervised approaches are particularly well suited for domains where expert annotation is costly and labels are subjective or ambiguous.

In this thesis we therefore focus on unsupervised approaches that learn from unlabelled data and are adaptable to different application domains. Our pipeline emphasises representation learning followed by low-dimensional embedding for exploration and labelling: (1) we encode images into meaningful high-dimensional representations, (2) we reduce dimensionality to create two-dimensional embeddings that preserve relationships between samples, and (3) we use those embeddings to guide labelling and analysis. Image-retrieval-style encoders compress information effectively but do not necessarily preserve global or semantic relationships unless trained with metric-learning objectives (e.g., triplet loss). Combining encoding with dimensionality reduction in a single labelling workflow proved faster and more useful for dataset curation than more disjointed alternatives—an advantage for remote sensing practitioners working with limited annotation budgets. A shortcoming of many existing labelling and segmentation approaches is insufficient attention to geographic context. Segments should reflect not only the object of interest but also the local neighbourhood and surrounding materials; without that context the semantic richness of remote-sensing datasets is lost. To address this, we propose and evaluate a pipeline that integrates image segmentation with graph neural networks (GNNs) to encode local geographic relationships. Our method constructs graph representations of segmented regions and applies a bipartite-graph matching scheme to align and compare local neighbourhoods across scenes. We further introduce a novel procedure to quantify the encoding capacity of GNNs with respect to local geographic context and demonstrate how this improves downstream labelling and variation assessment. Finally, we position our contributions with respect to the generalisability–accuracy trade-off common in the literature. Prior work often constrains training and evaluation to small, local regions because of data complexity and domain variation. By combining remote-sensing preprocessing, modern representation learning, and graph-based geographic encoding in a unified pipeline, we show the approach can be trained on localised regions yet generalise more robustly to larger, continental-scale areas for downstream classification and analysis.

This is arranged into the following chapters:

Chapter 2 Background This chapter initially introduces the problem statement from an in-

dustrial view. It covers the requirements and challenges faced by a Hydrographics office and the products they release, with a small historical background review of the UKHO and its motivations for remote sensing applications. The chapter also covers the basics of remote sensing technologies and apparatus with common applications and techniques. Also detailed is a brief understanding of remote sensing images from a physics and geology perspective. Lastly, common computer science techniques are detailed that are relevant to the thesis.

Chapter 3 Literature review This chapter provides an analysis of existing pipelines and labelling frameworks for remote sensing images within the literature. With discussion extended to image retrieval methods and iterative annotation. Lastly is a discussion on visualisation of remote sensing images is provided.

Chapter 4 Problem Statement This chapter summarises the current limitations of existing work and the root causes. With discussion on ideal outcomes for a labelling tool and the benefits of a new labelling paradigm.

Chapter 5 Unsupervised Scene Sample Extraction In this chapter, we propose our first unsupervised framework for encoding and labelling satellite images. Smaller images are encoded to represent a scene and are projected into two dimensions for user labelling. Our tool presents a novel labelling environment and is tested against existing methods in a user study.

The work presented in this chapter was published [29] in Tulsi Patel, Mark W. Jones and Thomas Redfern, *Manifold Explorer: Satellite Image Labelling and Clustering Tool with Using Deep Convolutional Autoencoders*, **Algorithms** 16(10):469, 2023. `https://dx.doi.org/10.3390/a16100469`

Chapter 6 Segmented Label Extraction Given the need for more fine-grained labelling, we propose encoding segmentations within this new work. The unsupervised encoding is substituted for a pseudo-supervised pipeline utilising fuzzy clustering. We utilise CNNs and GCNs to process segments and present a better labelling experience. The CNN features rivalling state-of-the-art performance.

The work presented in this chapter is under review [30] in the Taylor and Francis journal Annals of GIS with the title *Leveraging Convolutional and Graph Networks for an Unsupervised Remote Sensing Labelling Tool*.

Chapter 7 Interdisciplinary Unsupervised Labelling In this chapter, we introduce spectral unmixing as an alternative to an algorithm commonly used in the remote sensing field. We make no changes to the CNN or GCN architectures. The resulting feature space for segmentation is proven superior.

Chapter 8 Conclusions and Future Works Finally, we provide retrospective commentary and remarks on the presented works in this thesis. This chapter also details any of the challenges and assumptions made, for future improvements of any of the works.

Chapter 2

Background

Contents

Remote sensing (RS) is a broad discipline that has wide-ranging applications across many disciplines. Even everyday products like digital maps and weather images—often seen as simple RGB pictures—are built on complex RS technologies and principles. This background chapter introduces that complexity and connects it to long-standing challenges in cartography. Because this thesis is partnered with the UK Hydrographic Office (UKHO), which produces global charts for maritime navigation, we use their perspective to highlight the motivations and difficulties of large-scale geographic data assimilation and the resulting need for well-labelled RS datasets. The chapter then outlines the essentials of RS—from satellites to UAVs—along with common applications, key datasets, and the field's roots in geography and physics. It concludes with an introduction to the computer-science methods used in the thesis, including AI, visualisation, and manifold learning. The aim here is to establish relevant background, leaving detailed discussion for the literature review chapter .

2.1 SOLAS, navigation charts and products

Hydrography is the science of measuring and describing the physical features of oceans, seas, and other bodies of water—particularly depth and seafloor configuration—to create navigational products that support maritime safety [31]. National Hydrographic Offices, appointed by their governments, are responsible for producing these official nautical charts so that states comply with the International Convention for the Safety of Life at Sea (SOLAS [32]). First introduced in 1914 following the Titanic disaster and regularly updated since, SOLAS sets minimum safety standards for merchant vessels. All IMO member states with coastlines must therefore ensure their waters are adequately surveyed and charted, although some delegate this responsibility to another state's charting authority. As a result, certain Hydrographic Offices operate and conduct surveys across multiple countries.

SOLAS outlines specific requirements for the carriage and use of nautical charts. Under SOLAS, a nautical chart or publication is defined as an officially issued chart, book, or database designed to meet the needs of marine navigation. All ships, subject to SOLAS, are required to carry the charts and publications necessary to plan, display, and monitor their voyage. These requirements may be met using traditional paper charts or via an Electronic Chart Display and Information System (ECDIS), provided a suitable backup is available—commonly a complete folio of paper charts or an approved secondary ECDIS system. SOLAS further requires that all charts and publications, whether paper or electronic, remain adequate and up to date. Therefore, Hydrographic Offices have to maintain charts and publications, continuously sourcing

and analysing relevant data, assess implications for products, and issue guidance to mariners about changes required to navigation products.

2.1.0.1 UK Hydrographic Office (UKHO)

The United Kingdom Hydrographic Office (UKHO), founded in 1795 by royal warrant, is one of the world's oldest national Hydrographic Offices. Its early activities focused on producing charts, sailing directions and tide tables for the Royal Navy, and by the mid-19th century it had published nearly 2,000 charts for public and naval use. Throughout the 20th century, the UKHO expanded its remit to include oceanographic and meteorological information, invested in purpose-built survey vessels such as HMS Vidal, and progressively digitised its data and production workflows. The introduction of Electronic Navigational Charts (ENCs) and Electronic Chart Display and Information Systems (ECDIS) in the early 2000s marked its transition into fully digital chart production. The UKHO acts not only as the United Kingdom's national charting authority but also as the principal charting authority for numerous other states that cannot independently meet their obligations under the International Convention for the Safety of Life at Sea (SOLAS). It currently fulfils charting responsibilities for 63 nations across the South Pacific, Caribbean, Indian Ocean and South Atlantic [33]. Its work in these regions includes satellite-derived bathymetry, compilation of ENCs, and support for navigation, environmental monitoring, coastal-change assessment and disaster-management planning. The UKHO also collaborates internationally on marine environmental and hazard-monitoring initiatives—for example, providing expertise for the development of ocean and weather monitoring stations in partnership with other maritime administrations [34, 35]. The organisation maintains long-standing links with the UK Ministry of Defence and provides products and support for naval operations. Admiralty charts and associated products are used across the Royal Navy's surface fleet and submarine service. Current research and development efforts include automated mine detection and other advanced maritime safety technologies [35].

2.1.1 UKHO Charting Products

The UKHO produces two principal chart formats: paper charts (Standard Navigational Charts, SNCs) and digital charts (Electronic Navigational Charts, ENCs). SNCs are printed charts designed to be updated through manual corrections, while Raster Navigational Charts (RNCs) are electronic images of SNCs used on compatible display systems. ENCs, in contrast, are vector-based digital charts intended for use within ECDIS. Their vector representation enables

scale-independent visualisation, interactive querying and efficient storage. The global charting industry is progressively transitioning toward digital products. Many national authorities, including the National Oceanic and Atmospheric Administration (NOAA) in the United States, are phasing out traditional paper and raster charting in favour of ENCs and print-on-demand services, with NOAA aiming to complete this transition by 2025.

2.1.1.1 UKHO Use of Remote Sensing and Machine Learning

To provide a reliable global charting service in accordance with SOLAS, the UKHO maintains extensive workflows for data collection, quality assurance, analysis and integration with existing chart products. Key factors such as coastline change, seabed morphology, shipwrecks, oceanographic conditions and shipping traffic require continual monitoring. However, systematic in-situ observations are logistically challenging and often prohibitively expensive, particularly in remote or wide-area environments. Satellite remote sensing therefore represents a critical data source for maintaining global maritime situational awareness. The scale and frequency of satellite imagery required for global chart maintenance create significant demands on interpretation and analysis. Manual inspection of such datasets is impractical; consequently, the UKHO has expanded its capabilities in data science and machine learning to support automated extraction of relevant features and accelerate chart updates. A major challenge, however, is the limited availability of large, diverse and high-quality labelled datasets for training and validating machine-learning models across varied marine environments [14, 36]. Although the UKHO has released labelled datasets to support community research, the labour-intensive nature of data annotation limits scalability [14, 36]. New methods that reduce the dependence on manual labelling are therefore essential for future operational workflows.

2.1.2 Summary

The UKHO is a global charting authority with responsibilities extending across navigation safety, environmental monitoring and maritime defence. Its adherence to SOLAS requires frequent updates to charts—often weekly—based on evolving coastal, environmental and navigational conditions. The organisation is increasingly reliant on satellite remote sensing to monitor these changes efficiently across large spatial scales. Given the limitations of manual interpretation and the need for timely global updates, advances in machine learning, automated feature extraction and more efficient data-labelling methodologies are of strategic importance to the UKHO.

2.2 Remote Sensing Technologies

Figure 2.1: Example of an image taken from the Sentinel 2 mission with bottom-of-atmosphere(Level 2A) preprocessing. This tile can also be found in the SWED dataset [3]. The location is 30UYC according to the Military Grid Reference System (MGRS). We have composited this image using true-colour correction on the visible light bands of red, green, and blue. The spatial resolution is $10m$ with total coverage of $109800m^2$.

Remote sensing technologies are defined primarily by the platforms on which their sensors are deployed. These platforms fall into two broad categories: airborne and spaceborne. Airborne systems include unmanned aerial vehicles (UAVs), crewed aircraft, balloons, and other unmanned aerial systems (UAS). These platforms collect data from within the Earth's atmo-

Figure 2.2: Example of a Maltese cross configuration with 5 cameras. Four oblique cameras positions around a nadir. The example depicts the configuration used by TrackAir Midas System. Image sourced from [4].

sphere and typically operate at relatively low altitudes. Spaceborne systems are satellite-based and operate from orbital altitudes. They range from large, long-term missions such as Landsat and Sentinel to small, commercially produced CubeSats that use off-the-shelf components. Figure 2.1 shows an example image from the Sentinel-2 mission. Across both airborne and spaceborne platforms, four sensor characteristics—spatial, spectral, radiometric, and temporal resolution—govern the type and quality of data acquired:

- **Spatial resolution** (Ground Sampling Distance, GSD) refers to the physical area represented by each pixel. Airborne systems typically achieve resolutions of 5–25 cm, UAVs 1–5 cm, whereas satellite sensors range from approximately 0.3 m to 300 m per pixel. Higher platforms provide coarser resolution but much greater ground coverage. Satellites can image tens of kilometres per pass, while airborne systems typically cover 100 m to 1 km per swath.

- **Spectral resolution** denotes the range and number of electromagnetic wavelengths measured by a sensor. Different systems may capture broad multispectral bands or narrow hyperspectral channels, depending on the sensing apparatus installed.

- **Radiometric resolution** measures a sensor's sensitivity to differences in signal intensity, often expressed in bits (e.g., 8-bit or 16-bit). Higher radiometric resolution allows finer discrimination between subtle reflectance differences.

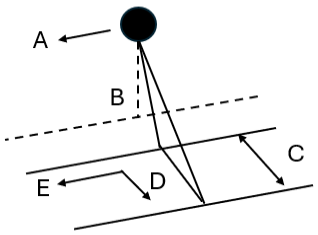
- **Temporal resolution** is the revisit time—the frequency with which a platform can acquire data over the same location. Airborne systems can be retasked to revisit within

hours or minutes, while satellites are constrained by orbital paths and typically revisit at multi-day intervals. Temporal resolution can be improved using satellite constellations or satellite trains, where multiple satellites cover the same area in coordinated succession.

The geometry of image acquisition is also influenced by platform altitude. Sensors may be oriented nadir (pointing directly downward) or oblique (typically 40–45° from vertical). UAVs and other low-altitude platforms often use a Maltese-cross configuration, with four oblique sensors surrounding a central nadir sensor, to improve coverage of vertical structures in urban environments (Figure 2.2). Spaceborne systems usually employ nadir-viewing configurations due to cost, complexity, and reduced suitability of oblique angles for high-altitude imaging.

2.3 Remote Sensing Apparatus

Figure 2.3: Example of a sensor's flight path and associated terminology. A is the flight path. B is the nadir. C is the swath. D is the ground sensing distance (GSD) and E is the azimuth.



Remote sensing instruments can be classified into two modalities: active and passive sensors.

- Active sensors emit electromagnetic energy and measure the return signal reflected from the Earth's surface. Common examples include radar and lidar, which provide information on surface structure and elevation.
- Passive sensors record naturally emitted or reflected radiation, typically from sunlight. Multispectral and hyperspectral imagers are the most common passive instruments, capturing reflected light across multiple wavelength bands.

Several geometric concepts apply across all platforms and sensor types (Figure 2.3). The flight path or orbit is denoted by (A), and the area directly beneath the sensor is the nadir

(B). The field of view (FOV) or swath width is represented by (C). The across-track direction (D), divided by pixel count, defines the spatial resolution or GSD, while the along-track (E) represents the azimuth.

2.3.1 Radar

Within remote sensing radar refers to synthetic aperture radar(SAR). Conventional systems coined radar is for detection an object and the range of the object allowing for the detection of the presence, speed, direction and type of object from a near stationary view. SAR is utilised to capture information from a moving platform, which is more common for RS applications.

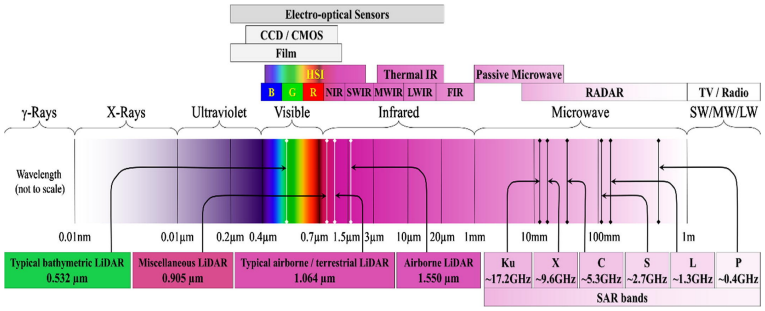


Figure 2.4: Typical range of wavelengths utilised in different apparatus. [1]

Radar emits and records information form the lowest frequency, energy and longest wavelengths from the electro-magnetic spectrum, the K,X,C,L and P bands, see Figure 2.4. These wavelengths can penetrate through different substrates. The X band reflects off leaves and canopies, C band penetrates to small branches, L penetrates to the trunk, and P penetrates to the trunk or ground. Minimal necessary accommodations are needed for cloud or weather coverage, in addition to illumination, for an active sensor. SAR is sensitive to rough surfaces, dielectric materials and wave polarisation and frequency. Rough surfaces cause diffuse scattering, reflecting energy in multiple directions including the sensor, resulting in higher backscatter. Dielectric materials, are poor conductors of electricity but can store electrical charges, such as wet surfaces can reflect stronger than dry materials.

SAR apparatus record in multiple different polarisation's. With difference polarisation's recorded certain characteristics of the observed scene and materials can be defined; structure, orientation and environmental conditions. For example linear direction of material, buildings, reflect in the same polar orientation. Randomly scattered material will be recorded in multiple polarisation's, depolarising the original scattered signal. In order to reserve the ex-

tra context SAR systems emit and record in certain polarisations. Single-polarisation systems, "single-pol", either emit horizontally or vertically and record in the same direction, Horizontal-Horizontal(HH) or Vertical-Vertical (VV). Dual-pol systems can emit and receive in different combinations of polarisation , HH and HV or VH and VV. Lastly quad-pol systems emit with twice the frequency to capture all combinations. Quad-pol can cause interference between the incoming echo, reflected values, therefore quai-quad-pole modes can be utilised. Quasi systems operate HH and HV modes in lower frequencies, for better surface detail in urban or rough terrain and VH and VV in the higher frequencies, for better penetration in forestry and crop structure.

2.3.2 Lidar

Another active sensing apparatus is Lidar. Lidar can be in theory utilised with any form of wavelength in the electromagnetic spectrum. Most common frequencies for Lidar are found between the 1-2μm wavelength intensities. Lasers have a high penetration through materials and commonly acquire the ground footprint. Depending the distance to the ground the laser provides a small or large circular area. As most Lidar systems are single, fixed wavelength, the time taken for detecting the return signal is more instrumental. The first returned signal denotes more elevated material and the last the lowest elevation material that can be detected. The return from Lidar systems create a point cloud, from multiple return signals depended on scatter and elevation of the target, which can be used to map elevation. Within airborne systems a more complex set of points are formed when multiple sensors are used in a Maltese formation as footprints can overlap. With spacebourne platforms a grid-like point-cloud is created based along the along-track. More recently multispectral lidar systems are being researched to incorporate multiple emitted spectra [37].

2.3.3 Multi and Hyper Spectral

Spectral imaging cameras that cover the mid-range of the electromagnetic spectrum are a form of passive sensing apparatus. They utilise the natural radiation provided by solar illumination. When illumination interacts with a material, the resulting reflected radiance wavelengths are altered according to the material's properties, enabling the delineation of the material's components. This is analogous to how humans distinguish vegetation from concrete when given a known library of spectra. The difference between the illumination source and the sensed intensity across multiple frequencies underpins the utility of spectral imaging. The more dis-

criminative a frequency is, the more information can be obtained. Multispectral images are typically recorded across 2–12 bands, whereas hyperspectral images comprise hundreds of bands.

The nature of passive sensing introduces several practical considerations related to the absorption and scattering of light. The first involves conditions within the top-of-atmosphere (TOA). Before solar illumination reaches the Earth's surface, it can be modulated by atmospheric conditions, including water vapour and aerosols. Additional illumination can also be scattered within the TOA toward the sensor, which is more prominent in spaceborne systems. This scattered light can be superimposed on light returning from a scene and is known as path radiance. Solar radiation is often scattered in the blue portion of the electromagnetic spectrum, producing diffuse illumination.

A second set of considerations relates to the bottom-of-atmosphere (BOA) scene. Here, illumination may scatter from nearby materials, affecting the final return spectra, or be occluded by shadows from clouds or topography. Finally, along the flight path, changing atmospheric conditions or variations in path radiance can alter the sensed signal as the apparatus moves.

2.4 Overarching applications of Remote Sensing Products

For many of the end-use cases, there are several main categories involved in extracting pertinent information from remote sensing product. From the perspective of building a labelling tool or aid it is important to understand the tasks that depend on labelled data. By identifying these end goals, we can more precisely frame what it means to label satellite data in a meaningful way. Each overarching application has a varying degree of reliance on labelled data. Although multiple methodologies have been explored over the years, which overlap and are discussed later, we look to summarise the overarching applications as done so by [38].

2.4.1 Target Recognition

Target recognition is a fundamental overarching application across many remote sensing tasks. Targets refer to features or materials within a scene that are of interest and need to be highlighted or extracted. Target recognition typically exploits spatial information, such as texture, often in combination with spectral signatures. This approach generally requires some a-priori knowledge of the targets to be identified. Examples include military and defence applications, such as infantry vehicle detection using SAR imagery [39] and ship detection [40] using multi-

spectral data. Similar approaches are used in civilian applications for the classification of crop disease, urban areas, water bodies, and many others [41–43]. While classification is a form of recognition that aims to characterise and identify materials, there is a related branch concerned with primary detection. Primary detection focuses on identifying changes or the evolution of features without necessarily characterising the material itself [44].

2.4.2 Anomaly Detection

Anomaly detection, aims to locate and identify uncommon features without needing a-priori information of what those anomalies look like. This application utilises the information from the scene to find differences and anomalies. Anomalous information is characterised by its deviation from background clutter, it is usually a low quantity of pixels differing from the surrounding pixel spectra [45]. Anomalous data can be a singular pixel, multiple pixels or even a variety of features so long they have a deviation from the background feature set. Use cases for such an application are military camouflage detection, silos, rooftops, vehicles on bridges or even large rocks in a grassy field [46, 47].

2.4.3 Background Characterisation

The goal of background characterisation is to find something other than spatially isolated features. The background encompasses an entire scene for analysis and identification. This can be seen in land, ocean and atmosphere domains; terrain categorisation, bathymetry or water vapour and aerosol identification [48–51]. Unlike target recognition and anomaly detection, the objective here is not to extract specific objects, but to model and label the broader environmental composition of the scene.

2.5 Datasets and Satellites

When considering the quantity of data, satellite platforms are the most viable due to the nature of continuous sensing. While airborne imagery is superior in resolution quality, the main contributors to open-source data are the vast data banks of satellite images. Sensing technology introduces variability between spatial resolutions, where airborne products are superior to spaceborne products. Higher spatial resolution increases the accuracy possible as more targets or objects of interest can be defined with a consistent border.

Name	Funding, operator	Launch	Country	Constellation	Sensor	GSD range (m)	Swath width (km)	Revisit time (day)
Ikonos-2	Commercial	1999	USA	Single	PAN 4 MS	0.8 × 0.8 3.2 × 3.2	11.3	3
QuickBird-2	Commercial	2001	USA	Single	PAN 4 MS	0.7 × 0.7 2.6 × 2.6	16.8–18	1–3.5
RapidEye	Commercial	2008	Germany	Five	5 MS	6.5 × 6.5	77	1–5.5
Pleiades 1	Commercial, government and private partnership	2011 2012	France	Dual	PAN 4 MS	0.5 × 0.5 2 × 2	20	1
SPOT 6	Commercial	2012	France	Dual	PAN 4 MS	1.5 × 1.5 6 × 6	60	1–5
Landsat-8	Government	2013	USA	Single	PAN 11 MS	15 × 15 30 × 30	185	16
SkySat	Commercial	2013 2014 2015	USA	2 (2014) 3 (2015) 24 full	PAN video PAN 4 MS	1.1 × 1.1 0.9 × 0.9 2 × 2	2 × 1 8	0.5 (2015) 0.12 (2017)
WorldView-3	Commercial, government and private partnership	2014	USA	Single	PAN 8 MS 8 MS (SWIR) 12 MS	0.3 × 0.3 1.2 × 1.2 3.7 × 3.7 30 × 30	13.1	1–4.5
Planet Labs	Commercial	2014 2015	USA	Flock of sats. (100+)	PAN 3 MS	3 × 3 5 × 5	Unknown	Unknown
DMC-3	Commercial	2015	UK	Triple	PAN 4 MS	1 × 1 4 × 4	23	1
Sentinel-2	Government	2015 2016	EU	Dual	13 MS 20 × 20	10 × 10 60 × 60	290	10 5 (dual)
Sentinel-3	Government	2015 2017	EU	Dual (triple planned)	21 MS 11 MS (IR) 1000 × 1000	300 × 300 500 × 500 1000 × 1000	1270 1420 750 (nadir)	0.25
Terra	Government	1999	USA Japan Canada	Single	14 MS (IR) 36 HSI	15 × 15 30 × 30 90 × 90 250 × 250	60 500 × 500 1000 × 1000	16
Aqua	Government	2002	USA	Single, part of A-Train	36 HSI	250 × 250 500 × 500 1000 × 1000	2330	1–2
EnMAP	Government	2017	Germany	Single	232 HSI	30 × 30	30	4
ICESat	Government	2003 2018	USA	Single	2 HSI 1 HSI (9-beam)	70 (footprint) 10 (footprint)	N/A	8 N/A
Envisat	Government	2002	EU	Single, tandem with ERS-2	C-band SAR	28 × 28 150 × 150 950 × 980	5 100 400	35 (orbit repetition)
RADARSAT-2	Government and private partnership	2007	Canada	Single	C-band SAR	3 × 3 100 × 100	20 500	24 (orbit repetition)
TerraSAR-X	Government	2007	Germany	Single	X-band SAR	1 × 1 16 × 16	5 × 10 1500 × 100	11
TanDEM-X	Government and private partnership	2010	Germany	Single, tandem with TerraSAR-X	X-band SAR	1 × 1 3 × 3 16 × 16	5 × 10 1500 × 30 1500 × 100	11
Sentinel-1	Government	2014 2016	EU	Dual	C-band SAR	5 × 5 5 × 20 25 × 40	80 250 400	12 6 (dual)

Table 2.1: Satellites taken from [1].

Consideration needs to be made regarding the satellite imagery that is available for processing. Table 2.1 lists typical satellites utilised in current datasets. Many satellites have different sensors, spatial resolutions and revisit times throughout the years. In particular, interest is in the Sentinel missions as they are government-funded and open source. They have the highest number of bands without compromising too much of their GSD.

There are two cross-modal operations between airborne and satellite imagery. Firstly, the study of pan-sharpening images where higher resolution airborne images can be matched to corresponding satellite images to upscale the later product. Satellite imagery itself is also conducive to upscaling between multiple satellite bands or missions, called super-resolution. Secondly, the ability to implement applications on satellite imagery and verify on higher resolution data.

Dataset construction and application pose a vast problem in remote sensing. Multiple avenues for collection can be undertaken for any domain-specific application. Extracting large amounts of data for generalisability in models should also be a consideration. Multiple features must be present and have multiple variations. This ideology can be encompassed in interclass and intraclass variation within classification.

The interclass variation relates to a set of multiple features; the more subtle the differences between the features, the lower the interclass dissimilarity. This variation can be seen in bare land and deserts, where texture and wavelengths are similar. To create robust real-world applications, the variance between similar features must be explored within a dataset for testing. This is also exacerbated by including more target features, which involves more sampling for lower interclass dissimilarity. Intraclass variation is the consideration of each feature's different orientation, imaging conditions or temporal conditions, to name a few. To account for these factors, datasets must be more extensive in quantity and quality. However, large datasets are expensive to curate, with expert labelling being expensive both in time and cost.

Dataset	Bands	Spatial Resolution (m)	Source	Classes (Examples)	Number of Images
UC-Merced [52]	R-G-B	0.3	United States Geological Survey National Maps	21 (Forest, Beach, Harbor)	2100
Brazilian Cerrado-Savanna Scenes [53]	R-G-NIR	5	RapidEye satellite	4 (agriculture, arboreal vegetation, etc.)	1311
Brazilian Coffee Scenes [54]	R-G-NIR	10	SPOT 5	2 (coffee and non-coffee)	2876
WHU-RS19 [55]	R-G-B	0.5	Google Earth	7 (forest, river, etc.)	1005
RSSCN7 [56]	R-G-B	1-8	Google Earth	7 (farmland, industrial, rivers, etc.)	2800
AID [15]	R-G-B	0.5-8	Google Earth	12 (park, agriculture, etc.)	10000
SIRI-WHU [57]	R-G-B	2	Google Earth	12 (agriculture, Commercial, Harbour)	2400
NWPU VHR-10 [58]	R-G-B	0.5–2	Google Earth and Airbourne	10 (airplane, tank, etc.)	715
SZTAKI-INRIA building [59]	R-G-B	0.3-1.5	Google Earth	2 (building, not building)	9
UCAS-AOD [60]	R-G-B	0.5	Google Earth	3 (airplane, car, background)	910
RSOD-Dataset [61]	R-G-B	0.3-3	Google Earth and Tianditu	4 (aircraft, playground, overpass, oil-tank)	2326
Vaihingen [62]	R-G-B	0.09	airborne image	6 (building, tree, car, etc.)	33
Potsdam [62]	R-G-B-NIR	0.05	airborne image	6 (building, tree, car, etc.)	38
PatternNet Images [63]	R-G-B	0.062-4.693	Google Earth	38(airplane, river, closed road)	800

Table 2.2: Example of RS datasets

Another consideration for any application is the quality and quantity of labelled images, which is sparse for all forms of RS platforms relative to the data produced. Table 2.2 shows some standard datasets utilised and their respective sources, which is a non-exhaustive list. For more datasets produced up to 2021 see Figure 2.5 [5]. Dataset creation and utility are mainly found in semantic segmentation, classification and object detection applications. Classification can be utilised for scene-based, object-based or pixel-based classification, with more of a focus on the latter [64].

Datasets based on satellite imagery commonly have singular use cases. For example, the target recognition of water edges (SWED), marine debris and crop mapping [14, 65, 66]. Singular-use datasets such as LandsatSCD or MSCDUnet are also utilised for change detection [67, 68]. As a general trend, images that contain multi-classifications are labelled for crop mapping or specific land use. Diverging away from airborne imagery, most datasets have lower

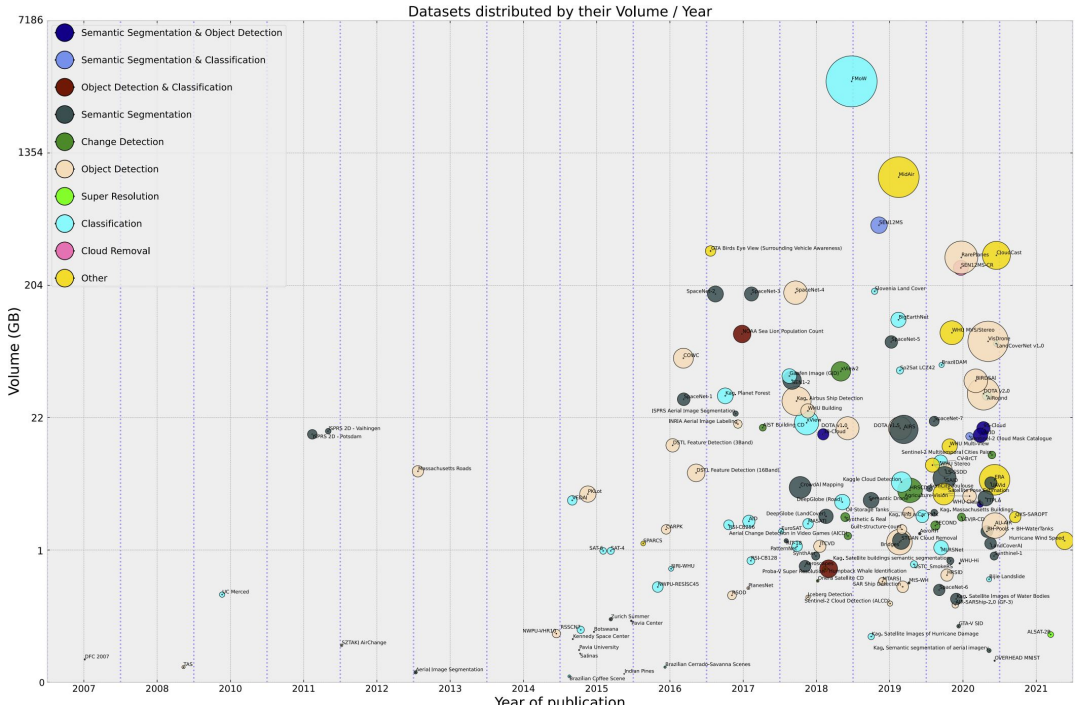


Figure 2.5: This figure and caption are from Schmitt et al. page 2 [5]."Evolution of remote sensing datasets dedicated to machine learning tasks. Since dataset "size" being a hard-to-define measure, it is represented in two ways: The vertical axis relates to the actual data volume, while the circle size relates to the number of spatial pixels covered by the dataset. This way, size is connected to both the spatial dimension as well as the overall information content in terms of implicit features such as resolution, sensors modalities, numbers of bands/channels etc."

spatial resolutions with higher spectral and temporal resolutions. The difficulty of acquiring labels for any product means that only specific geographical locations are selected for each use case; therefore, organising and obtaining data from multiple locations can be costly.

2.6 Remote sensing: Physics and Geology definition

The arsenal of machine learning techniques and methods that are utilised for imaging contain profound challenges and results. Most methods can navigate between multiple different domains such as, CCTV footage, pose estimation or other image datasets. Understanding each domain's subtle complexities can add constructive analysis to the results and formulation of the problem domain. Within this section, we look at the problem domain via the lens of remote sensing to provide a more concrete foundation. Secondly we highlight a few methodologies that are utilised for respective problems.

For the unique properties of remote sensing data, specifically multispectral and hyperspectral data, processes and methods have been developed that are specific to the RS field. Notably, there is a need to compare the spectra libraries to sensed mixed return signals. Spectra libraries are the recorded wavelengths of materials under specific illumination conditions. Many factors affect these return signals, such as sun angle, atmospheric properties, material pigments, orientation, size, texture and spectral mixing. Early works looked towards creating radiative transfer models that took the end spectral return to estimate factors. For example, MODTRAN is a well-developed tool for predicting and analysing atmospheric distortions [69] or PROSPECT-D, which models leaf optical properties [70]. Radiative transfer models were born between the joint domains of physics and remote sensing. Although informative for particular use cases and analyses, most are founded outside the computer science domain. We can see the use of genetic algorithms within this field [71].

Including such preprocessing for atmospheric consideration is essential for creating any datasets. Datasets utilise only optimal images, that can be acquired given optimal sensing conditions. For airborne imagery external conditional variation can be mitigated by only launching for data capture at the correct time. In spacebourne imagery selective data can be accumulated where unwanted conditions are absent or minimal, the ease of which is dictated by the temporal resolution. With extra radiometric correction normally applied to spacebourne images, these can be seen within "levels" of each product produced by a satellite. For an example, see figure 2.6, for Sentinel 2 levels of products and associated preprocessing. In which top-of-atmosphere (TOA) reflectance is calculated utilising algorithms similar to MODTRAN [6]. TOA, also known as the exosphere, is the outermost layer of the Earths atmosphere. There are level two products that convert bottom of atmosphere(BOA), ground level, from top of atmosphere for Sentinel 2 products, in which cirrus cloud correction is undertaken and water vapour retrieval.

Once we have suitable BOA images the main challenges are finding endmembers and their abundances if needed. Endmemebers are the set of distinct, macroscopic, materials that are being sensed, such as, soil, water or rock minerals. This definition distills the whole idea of remote sensing into a simple terminology, built upon this we can define multiple different RS problems. The first of which is the inclusion of the term abundance. A singular pixel can include more than one distinct material, for one of two reasons. The spatial resolution of the sensor is too low, consider the area covered by a one, or ten, metre squared area. Secondly independent of spatial resolution the occurrence of substances can combine into a homogeneous

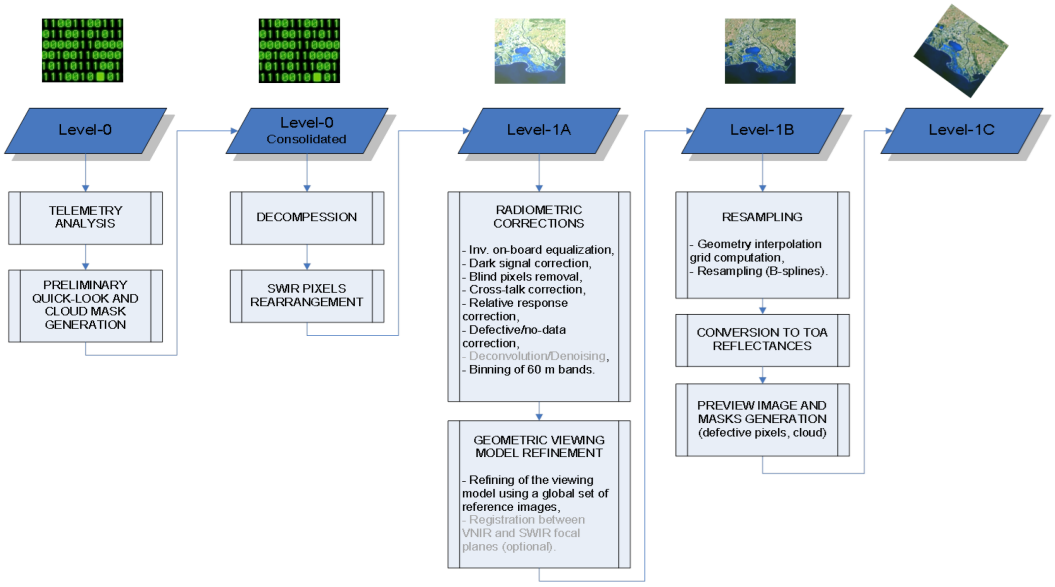


Figure 2.6: Sentinel 2 Preprocessing, [6]

mixture, for example the partition of sand and sea at a beach. The abundance or fractional abundance is the proportion of endmembers material sensed within a singular pixel. These two terms are from the study of spectral unmixing to decompose a sensed spectra into its proportional constituent parts [72].

Where singular incident radiation, source of illumination, hits a singular material or substrate the surface area of the sensed area could be considered proportional. In short there is a direct relation between the sensed abundances and the area coverage. In contrast there can be scattering between materials where the radiation is reflected and absorbed by multiple materials before being sensed therefore a non-linear relationship. These two modalities of mixing are both reflected in linear and non linear unmixing models.

Determining endmember pixels for comparison is also a crucial component when applying any subsequent unmixing models. The endemebers are usually collected via expert knowledge in the domain. Interactively selecting obvious pixels that are pure to the scene and analysing before collecting more endemebers to the selection of chosen pixels. It is impossible to ascertain a pixel which purely consists of one material. Each endmember is also dependent on the scenes temporal factors, the sun azimuth, topography etc. Given that topography is always inconsistent both on a large scale and smaller scale, mountains or tree canopy for instance, the ability to get endmember spectra that have not been refracted off multiple elevated topographical material is rare. In addition there is also variability in the temporal scale via the

aforementioned factors, seasonal and diuranal cycles. A large amount of endmember spectra need to be evaluated to conceive and all encompassing library of reference spectra. There are multiple algorithms that help ascertain the most pure pixels, pixel purity index, N-FINDR or convex hull analysis [73–77].

For any pure pixel considered an endmember a complex process can be undertaken. Utilising only the spectral values we can compare how close certain pixels are by utilising similarity or distance measures. Within remote sensing classical similarity measure such as Euclidean are still utilised however specialised versions have been produced. Spectral angle mapper (SAM) is such a comparative measure, where each spectral spectral signature is considered a vector and a comparison is made between the angle between each spectra [78]. SAM, whilst quite prominent, had one flaw by treating the spectra as a vector with absolute value unable to distinguish between positive and negative correlation. Another consequence was the inability to model the curve of the spectra, the shape formed when considering the difference between wavelength returns for a pixel. Addressing these concerns spectral correlation mapper was utilised based from the Pearson correlation coefficient [79]. Alternatively spectral information divergence models each spectra as a probability distribution so the variability between each band can be measured stochastically [80].

Due to the large size of areas in consideration for any given satellite product, referring back to table 2.1, where a large swath and and low pixel resolution produces millions of pixels in one image it is only natural to consider some form of reduction. Dimensionality reduction aims to retain the most pertinent information whilst reducing the over dimensionality, number of bands considered. This can expedite subsequent processing as algorithms are computationally faster on lower dimensional data. A common staple within machine learning is principle component analysis where information undergoes eigendecomposition to find orthogonal axis [81]. Whilst PCA finds the most informative information the resulting reductions do not assure the retention of information that can detect low probability objects and is sensitive to noise. Alternatively two very similar algorithms maximum noise fraction(MNF) and noise adjusted PCA aim to provide reduction with regards to noise.

In contrast to statistical reduction and spectral unmixing, remote sensing allows for informative pre-selection of bands based on the target criteria. As stated before each material will have some absorption characteristic per band resulting in different intensities. Utilising a-prori information of a particular material or substrate indices, ratios can be created. A popular example of an index is the normalised difference vegetation index (NDVI) for detection and extrac-

tion of vegetation health. Plants photosynthesis within the photo-synthetically active radiation spectral region absorbing and re-emitting drastically less in this range whilst reflecting nearly all higher energy wavelengths, reducing the damage that it incurs to the plant. NDVI utilises the near infrared bands, higher energy, and red light, photo-synthetically active, spectrums via the following $\frac{NIR-RED}{NIR+RED}$ [82], where NIR is the sensed value of the near infrared wavelength and RED is the sensed value of the red wavelength. The final output is ranged between -1 and 1 and can be thresholded for each particular region to highlight plant growth and health. When such ratios are applied to a singular image they have the ability to mitigate the effects of atmospheric conditions, sun angle and other factors as they are constant for the entire image. There are many more indicies that have been utilised; Anthyocyanins reflective index, Chlorophyll red edge, normalised difference water index, normalised snow index, marine cyanobacteria, bathymetry and many more [83–98].

2.7 Towards Machine Learning

The following sections will give a review of the most relevant classical techniques, sections 2.7.1 - 2.7.4, that provide a background to the thesis. In particular the clustering techniques described in section 2.7.4 are the most relevant to the works in this thesis. Discourse is mainly focused toward specific examples that are constrained to the remote sensing field. Reviews and surveys in this field cover a wide range of topics, we recommend the reader to the following surveys for a comprehensive background on all classical techniques: Asokan et al. for **analysing remote sensing images** [99](enhancement, feature extraction, segmentation, Fusion, Classification, Feature detection), Babbar et al. for **land use and land cover analysis techniques** [100](Region Based, Edge based, Pixel based, Thresholding, Supervised or Unsupervised feature extraction), Ouchra et al. for **classification techniques** [101](Supervised, Unsupervised, Object orientated, Pixel based, CNN) and Bagwari et al. for **segmentation techniques** [102](Edge-based, Thresholding, Region, Clustering, Deep-Learning). This survey covers several of the main classical techniques relevant to RS.

2.7.1 Edge-Based Feature Extraction

Edges within an image reveal homogeneous regions of similar items and the texture of a larger group of homogeneous areas. For each substrate or material detected, some boundary will exist where the spectral composition will change to another material. This computation is usually

restricted to where each region has a similar abundance per pixel. We can utilise simple convolutions with handmade filters to find the edges of these changes. For example, some methods utilise multiple passes of different filters to extract contours or edges. Each filter is orientated towards a cardinal direction, horizontal, vertical and diagonals, an example being Roberts cross [103]. Sobel, Prewitt, and Kirsch are well-known classical algorithms [104, 105]. Each method can be considered to be taking the derivative along the cardinal direction associated per filter. Therefore, when calculating the changes in the first-order derivation, any inclusion of noise can adversely affect the resulting edges. Considering RS images, these noises or natural imperfections are many and varied. Depending on the need for strong or weak edges, the resulting image can be further thresholded to single out different intensities. A Gaussian filter can be applied to smooth broken edges or remove imperfections beforehand. Canny produced a multistage method utilising smoothing before applying filters [106]. Alternatively, the second-order derivative can be utilised for less sensitivity to smaller intensity changes, as seen in Laplacian of Gaussian [107]. Liu et al utilised edge detection for land and water boundary detection [108]. Other works cover a broad range, looking to denoise and extract features or utility within image retrieval [3, 109–111].

We can find many more applications within the image processing domain: fuzzy, spatial-frequency, anisotropic or active contour-based edge detectors [112]. We have mainly noticed that contour-based applications receive attention for recent works regarding RS. Where previous methods are localised to the filter's region, active contour models look to find global discontinuous outlines. Examples can be seen for oil slick or coal fire detection [113, 114]. Wei et al. produce an initial contour and then apply an improved geometric active contour model to delineate the boundary between land and sea [115]. Active contour models work on an initial deformable curve optimised through an energy function. Parametric models utilise the parameters set as the driving force for optimising the curve. Whereas geometric models embed two-dimensional contour into a level set function of three of higher dimensions, then solve it for the zero level set [116].

2.7.2 Thresholding

Thresholding concerns the delineation between two categories, foreground and background. As simple as the definition is, its utility can be found across many different applications in remote sensing. As discussed in section 2.6, normalised indexes require a cut-off point. Sekertekin produced a survey denoting many other methods to threshold after utilising the normal

difference water index [117]. They reviewed 15 different methods in their study, of which we reference a few. One of the most well-known methods is Otsu's, where the optimal threshold value is calculated by maximising the between-class variance regarding a grey-level histogram [118]. A large discrepancy between the two peaks within the histogram is generally needed for an optimal result using the Otsu method. Given that noise and other factors can be detrimental to the result. Likewise, some similar histogram thresholding techniques included within the survey are inter-mode or minimum-mode [119]. Other methods focus on cross-entropy, either minimising or maximising via calculating probability distributions [117]. Commonly cited is the Kapur method for cross-entropy between the foreground and background [120].

Alternatively, multiple different methods imitating natural processes have also been used in thresholding. Pandley et al. use such thresholding methods for satellite image segmentation [121]. Notably genetic algorithms, particle swarm optimisation, artificial bee colony, Cuckoo search and their respective variants.

2.7.3 Region

Region segmentation of an image is an amalgamation of region growing, splitting, and merging techniques. Initial seed points are usually acquired based on predefined criteria before each seed is grown by considering neighbouring pixels. All pixels are assigned to a local seed point; therefore, each area of an image is segmented. The goal for each region is to have similar regions with the same area, texture, and colour. The boundary of a region can also be considered a descriptor, such as the length or curvature. Given the structure and descriptors of a region, they can also be split or merged depending on whether the conditions initially set are satisfied.

The watershed method and its variations are a commonly used image processing technique [122]. Its derivation is metaphorically analogous to an area's geological watershed, separating drainage basins. The initial processing is conducted on a greyscale image. High intensities are considered peaks, and low intensities are valleys. From the initial low points, water is flooded gradually until the peaks are covered in water. The segmentation is formed when two water sources meet at various stages of flooding. This method introduces unwanted variability to low-intensity noise, local minima, which can be initialised as a seed point. As seen before, noise can be countered with Gaussian blur; however, caution must be taken to avoid creating misaligned edges. The initial selection of the seed is a prominent problem within region-

growing techniques. Where algorithms provide too many segments due to poor initial seed selection is referred to as over-segmentation. While over-segmentation can accurately define and split an image into multiple homogeneous regions in real-world applications, some areas can be considered background and add unnecessary complexity when segmented [123]. The initial seeds can be carefully selected, known as markers, to guide the extraction process to improve performance. Genitha et al. utilise markers before applying watershed to extract marine vessels [124]. Alternate strategies can produce over-segmented images by applying a uniform grid of initial seed selection [125]. The spectral angle can be compared to evaluate each adjacent segment to threshold the merging of two segmentations [126].

2.7.4 Clustering

Clustering, unlike the region-based approach, considers the entirety of a scene and splits information based on each pixel, usually regardless of neighbouring information. Clustering provides an unsupervised way to split information based on some preset parameters. The shared parameter in most clustering algorithms is the number of clusters, such as K-means, or the threshold from which clusters are determined, such as hierarchical clustering approaches. Hierarchical and partitional comprise the two main categories within this field. Hierarchical clustering aims to compare the closest pixels iteratively, creating a bottom-up approach known as agglomerative or splitting in a top-down fashion known as divisive. Both methods can be modelled into a tree structure called a dendrogram, from which cuts can be made to select clusters. Santos et al. utilise a hierarchical approach in combination with self-organising maps (SOM) for primary clusters to identify land cover use [127]. The complete relationship means clusters can be easily identified, and data can be of any arbitrary shape, unfortunately requiring higher time complexity than other clustering algorithms. Ji et al. show the utility of divisive clustering when considering band selection for classification [128].

Comparatively, partitional clustering disregards modelling all points to each other and is generally faster computationally. The most well-known is K-means within the literature for clustering [129–132]. Its popularity can be attributed to its low time and computational cost. However, the use cases must consider the high sensitivity to outliers, the initial clusters parameter, and the trend of settling for local optimals. A fuzzy approach can be used where use cases require the modelling of abundances for each pixel. Fuzzy clustering proved a membership to each cluster represented with probability. Zhang et al. utilise fuzzy clustering with superpixel anchors for acceleration and deriving cluster labels [133]. Whilst fuzzy clustering is more re-

alistic, providing membership to each cluster has a cost in terms of time complexity and shares the tendency to be drawn into a local optimal.

Other methodologies in clustering focus on the density of the data to cluster. Density-based algorithms forgo the need for a predetermined set of clusters to find. Instead, they focus on the distance between points, define a threshold for a cut-off point, to derive final clusters. Estimations of the threshold are of increased importance and can be detrimental to finding when the clustering space is not even. Density-based spatial clustering of applications with noise (DBSCAN) has been utilised for marine trajectory and anomaly detection [134].

When the data can be converted to relationships between the points, be it geographically or in a custom modelled relationship, we can convert the data to a graph structure. The clustering techniques utilised on graphs are very computationally and time-consuming; however, they provide very high accuracy. Cai et al. show the benefits of graph modelling and graph spectral clustering for multiple datasets [135].

Finally, we come to models such as Gaussian mixture models (GMM) or Kohens self-organising maps (SOM), where each cluster is assumed to fit an underlying probability distribution. Models are built and fit on a subset of data to understand the underlying principle data. The model is then used on unseen data, in which the learnt probabilities are used to cluster data. Well-defined models provide suitable clustering results. Given each model's unique characteristic, the use case and utility can vary, as well as the cost of training. GMMs have been utilised for segmentation, anomaly detection and specific feature finding to reasonable accuracy [136–138]. SOMs likewise share the same use-cases [129, 139, 140].

2.8 Thesis Techniques

This following section introduce the background of techniques within the thesis including, autoencoders, attention mechanisms, graph neural networks, manifold learning and some background of image processing outside of the remote sensing domain.

2.9 Autoencoders

This section introduces multiple concepts; architecture, labelling and ground truth. Architecture refers to the layout of a model in reference to the layers how information is passed through the connections, from which the shape at each layer, output range and number of weights can be defined. Labelling is assigning a value to each pixel or image that uniquely belongs within

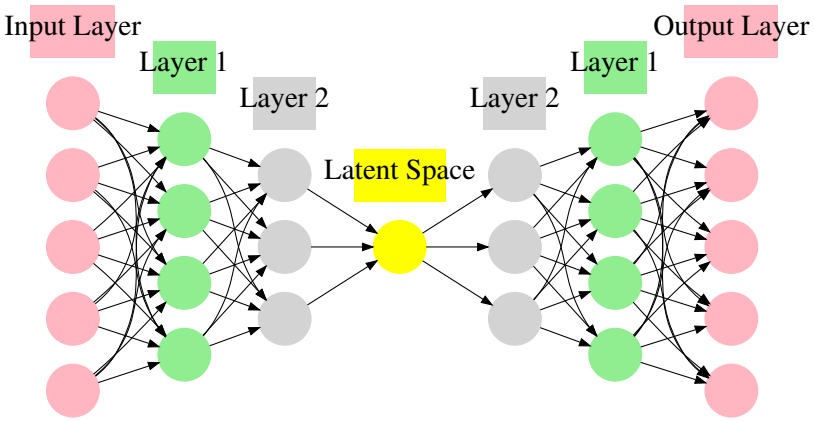


Figure 2.7: Example of an autoencoder with 2 hidden layers.

a category, these labels if accurate are referred to the ground truth. Architectures that utilise ground truth or labels within their loss function are considered supervised models, those that do not are unsupervised.

When considering a domain where labelled or ground truth data is hard to establish, which is needed to effectively train and extract features, we consider unsupervised approaches. Considering that remote sensing pertains to mapping the entire global surface of Earth ground truth data is at time impossible or unfeasible to establish. This is where autoencoder (AE), an architecture for both NN and CNNs, plays a role. AE can be utilised on unlabelled data as the cost can be calculated based solely on the input. The main cost function being a mapping of the difference/similarity between the input data and output of the model. Within an AE an encoder takes the input space and encodes it into a feature space and a decoder which aims to generate the original input back from the feature space. As such an AE is a generative model. As the encoder and decoder are simply the inverse of each other the following hold true for the weights, W, $W_y = W'_z = W$ [141]. That being said the weights of the decoder and encoder are not shared and non-linear. In a perfect solution the weights of both models, the encoder and decoder, would be the same but in practice they are always different. The goal is to minimise the error between the input and outputs and therefore is an approximation for most complex input functions. The utility of the AE lies in the latent feature space between the encoder and decoder that now represents the initial input given that the cost function is adequately satisfied. An AE is only one layer deep and therefore only provides lower level representation of the feature space. To obtain higher level features we append multiple layers onto both the encoder and decoder forming a stacked auto encoder (SAE).

2.10 Attention Mechanisms

Attention mechanisms were introduced in the domain of language translation and are a more recent addition to the Deep learning [142]. As the original utility for attention mechanisms was produced for language translation in an architecture called transformers we are going to utilise that example to explore attention. The transformer utilised a AE architecture with attention mechanisms in each encoder and decoder block. The main goal was to encode a language and decode into a translation. When considering a sentence in the input space there are multiple words that could affect how other words in a sentence are translated, an example being the grammatical gender of a word. In such instances the utility of encoding the relation between words becomes highly useful and a dependency. To solve this, attention mechanisms were developed. The attention mechanism could assign importance between the relationships between words and the amount of attention to apply to each relationship. There was also the ability to encode the attention of a word to itself. To generalise, attention mechanisms calculate the weighted sum of elements in the input space, where the weights are learnable and indicate the relevance of each input element. There are various types of attention mechanisms when calculating the weighted summation to name a few, additive, multiplicative and scaled dot-product. There is also the ability to allow for multiple attention heads, multi-head, that increases the different representation sub-spaces.

2.11 Graph Neural Networks

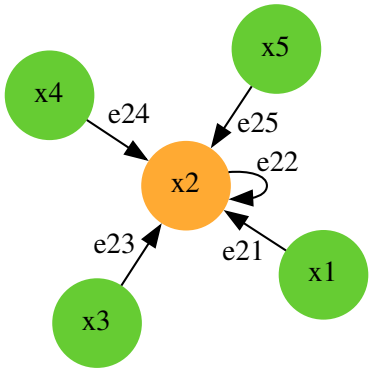


Figure 2.8: Message passing for node X_2.

Graph Neural Networks (GNN) are designed to work with graph structured data. A graph is defined by a set of nodes, or vertices, and edges. The nodes represent features and edges

represent the connection between features. Graphs can be very flexible in structure where each node can have an edge to every other node, no other node, or anything in between. They also allow for directed edges for connections that are not bi-directional. GNNs gained popularity due to their ability to capture complex relationships between features within a dataset and the ability to transfer the relationship into a model.

Message passing is the primary methodology by which GNNs learn information. Message passing refers to the interaction between nodes in a graph layer. Each node shares information with the connected nodes along the predefined edges. Much like the localised ability of a CNN, however the localised neighbourhood can be predefined to information that is not spatially close. This can also be seen by the adaptation of the convolutional operation to this irregular domain, which is known as neighbourhood aggregation or message passing:

$$x_i^k = \gamma^k(x_i^{k-1}, \oplus_{j \in N(i)} \phi^k(x_i^{(k-1)}, x_j^{(k-1)}, e_{j,i}))$$

Where x_i^k is the features of node i in layer $(k-1)$ with edges $e_{j,i}$. $\oplus$ denotes a differentiable function, for example, sum or max, and γ and ϕ represent differentiable functions such as MLPs.

GNN layers also can introduce features that both NNs and CNNs have utilised such as attention mechanisms. Attention within a new graph node X'_i, can be defined as such:

$$X'_i = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in N(i) \cup i} \alpha_{ij} W^k x_j\right)$$

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

$$\alpha_{ij} = \frac{\exp\left(a^T \text{LeakyReLU}\left([Wx_i \| Wx_j \| e_{ij}]\right)\right)}{\sum_{k \in N(i) \cup i} \exp\left(a^T \text{LeakyReLU}\left([Wx_i \| Wh_k \| e_{ik}]\right)\right)}$$

where e_{ik} is an edge between node i and connected node k. Attention in this form can allow for a each node to weight the importance of each surrounding or connected node by an edge.

2.12 Deep Learning Evolution within Image Processing

This section will provide a brief overview of the most common paradigms utilised within the thesis. The evolution and history or key points regarding these techniques will help readers

unfamiliar with these areas to gain a general understanding of the respective fields.

2.12.1 CNNs Architectures

As the output space of most DL methods is a highly complex hyperplane created by a complex function space, the valuable separation of samples in that space is difficult to find. Most architectures utilise further models or layers to find and optimise the separation between points. This training methodology has been highly effective as the boundary in the output space is known and can be further exploited in training to create definitive predictions. In 2012, Sutskever et al created "AlexNet", which utilised dropout regularisation and the rectified linear units (ReLU) [143]. They won the ImageNet Large Scale Visual Recognition Competition with their model [144]. Zhong et al. in a survey, explain the linkage between two further works that built upon AlexNet [145]. The first paper utilised AlexNet and showed that features can be re-utilised on different generic imaging tasks [146]. This was further expanded upon by Zhong et al by taking the sixth layer and applying a state vector machine (SVM) for classification on the hyperspectral plane [147]. The ability to represent features at a generic level can be approached by taking pre-trained models and extracting different generic features. In years to come many models have been released to approach image classification. "ResNet" a deep residual network introduced in 2016 by Zhang and Sun utilised skip connections. Skip connections allowed for earlier layers in a model to share information with non-adjacent layers deeper in the model. This allowed for learning on features combined with lower level feature representation of contours and edges with deeper-level features. More importantly, it addressed the vanishing gradient problem where deeper models suffered. Further work from Google Deep mind and Oxford University produced the VGG model (Visual geometry group) [148]. Their approach was to increase the depth of the network using small filters and strides; in doing so the model outperformed "AlexNet" and others in classification tasks across domains.

2.12.2 Attention Mechanisms

There has also been a rise of attention mechanisms applied to image processing. The human visual recognition system is the main motivation for attention mechanisms. The core of computer vision tasks is the ability to find the important aspects of an image in relation to one another with explainable decisions. SENet proposed by Hu et al. for channel based attention [149]. They developed a Squeeze and excitation block (SE), which models the interdependences between channels. By including SE blocks within the ResNet architecture they showed improved

results for scene classification. Attention mechanisms are computed on the activation maps of CNNs and therefore can encode global information which when considering the locality of CNNs is extremely valuable, subsequent layers of the CNN have more additional context of the image from different regions. A survey by Guo et al detail the advancements of attention mechanisms [150]. Most applications of attention mechanisms based on the application provide certain attention mechanisms within image processing, Spatial, Temporal, Channel, branch or combination. Each task within computer vision requires a special version of an attention mechanism. Guo also concludes the different applications in which attention can be utilised. Channel attention focuses on "what to pay attention to", which is a good choice for image classification. For dense prediction tasks spatial attention is required, "where to pay attention". Temporal attention is suited for time series data, which are adopted into a recurrent neural network(RNN) or long-term short-term memory (LSTM) networks. As additional attention layers are added to a model the computational complexity of the model also increases.

2.12.3 Graph Neural Networks Applications

Graphs have been prevalent in research before they were applied to NN domain. Graphs can model information within irregular non-euclidean domain. Graph structures were first applied to neural networks by Sperduti and Starita [151]. The initial applications fell into the category of recurrent neural networks. As CNN's grew popular, so did its influence in other domains, including graphs. The idea of utilising convolution on graph structures was born (ConvGNN). There were two main streams to ConvGNNs: spectral and spatial methods. Spectral methods utilised the Eigen decomposition of the graph's Laplacian matrix. Eigendecomposition allows for an understanding of the clusters and underlying structure. Alternatively, spatial methods work on local neighbourhoods where only particular neighbours are taken into account, this method is much more adept in flexibility, generality and efficiency [152]. A survey by Wu et al focuses on two more evolutions of graph NN, the graph autoencoder and Spatial-Temporal graphs. Graph autoencoders, like CNNs, encode graph information into a latent vector and reconstruct from the encoding. Finally, spatial temporal GNNs aim to find patterns within both those features, utility of which is seen in various applications including, traffic speed forecasting, driver manoeuvre anticipation and human action recognition [153].

2.13 Manifold Learning

Manifold learning is finding structure of high dimensional data. Also known as non-linear dimensionality reduction, finds structure by projecting to a lower dimensional space. Dimension reduction aims to describe the geometric shape of the data in order to interpret or visualise. As human interpretation is more efficient within a two or three dimensional space these methods also serve as a way to compress data into lower dimensions.

A manifold is a mathematical space the in local regions looks like a Euclidean space $\mathbb{R}$ but may have more complicated global structure. For example the Earth is spherical however on local charts or projections are mapped to a flat Euclidean space. Mathematically a manifold M is defined by these factors [154]:

- For each point $p \in M$, there should be a small open neighbourhood $U \subset M$ of p and a map φ. φ maps points in U to an open set in $\mathbb{R}^d$ which is bijective and smooth. This pair (U, φ) is called a chart. The inverse map $\varphi^{-1} : \mathbb{R} \to M$ is called a local coordinate.
- For two overlapping charts (U, φ) and (V, ϕ) the transition between one chart via the manifold should be smooth $\phi \circ \varphi^{-1} : \varphi(U \cap V) \to \phi(U \cap V)$

A manifold by the above definition has smooth transitions, has distinct points that can be separated by neighbourhoods (Hausdorff) and is second countable (has a countable base for the topology). The last two, Hausdorff and second-countable are satisfied by most spaces in statistics or manifold learning. The simplest manifold is one that only requires one global chart, consider a 2D surface rolled in 3D space. On the other hand, a sphere or torus, also a 2-manifold, would require multiple charts. For example Earth's surface has Mercator projections, for equatorial regions and navigation, and polar stereographic for polar regions. Different charts can also be utilised to describe the same region therefore care must be taken when comparing results from different methods or samples.

Embedding is how a manifold M is placed into another space N. An embedding $F : M \to N$ is injective, smooth and has a smooth inverse. If $M \subset \mathbb{R}^{\mathbb{D}}$ then $F(M) \subset \mathbb{R}^{>}$ where m is usually less than D.

Distances calculations along a manifold are also critical, there may be a need to restrict the distances from all of $\mathbb{R}^{\mathbb{D}}$ to the manifold $M \subset \mathbb{R}^D$. Consider the Earth examples again, a sphere, mathematically in $\mathbb{R}^{\mathbb{D}}$ can be traversed in a Euclidean straight line between two points,$||x_1 - x_2||$ however this may cut through the ground. Alternatively restrict to $M \subset \mathbb{R}^{\mathbb{D}}$

for physically valid configurations, in this way traversing valid points within M from x_1 to x_2. The shortest length of such a path is called the geodesic distance. Geodesics is a foundational concept in Riemannian geometry.

For each point $p \in M$ understanding how a smooth map affects each point is considered the differential. The differential dF_p is a linear map that, it maps the tangent vector at p in the manifold M to a tangent vector $F(p)$ in manifold N.

$$dF_p : T_pM \to T_{f(p)}N$$

dF_p can be considered the localised behaviour of the mapping function F near the point p. This in coordinates is a matrix, the Jacobian of F, $dimN \times dimM$. For any two Riemannian manifolds, where there exists a choice of inner product at each tangent space at p, a perfect embedding can be described. This embedding is a isometry it preserves all geometric quantities for all tangent vectors.

$$\langle \upsilon_1, \upsilon_2 \rangle_{g(p)} = \langle dF_p(\upsilon_1, dF_p(\upsilon_2) \rangle_{h(F(p))}$$

Where two tangent vectors υ_1 and υ_2 at points $p \in M$ measured using g is equal to the inner product of their images under the differential dF_p at $F(p) \in N$ measured using h. g and h are measures of distance, angles, length, volume, and curvature. If and only if the map of F preserves the inner product for all points $p \in M$ and all vectors $\upsilon_1, \upsilon_2 \in T_pM$ is defined F as an isometry. In mathematical theory this has been proven however no known practical implementation is capable of an isometric embedding [155].

Nearly all implementations begin with the notion of constructing an graph by calculating localised neighbourhoods. The choice of neighbourhood, N_i, for each point x_i, is an encoding of the local topological information. The neighbours of x_i can be calculated in two usual ways, radius neighbours or k-nearest neighbours (k-NN). The radius neighbours of point x_i is point x_j iff $|x_i - x_j| \leq r$ where r is used to control the scale. In radius graph building the density of points is crucial and directly related to r. The K-NN graph defines a neighbour x_j iff x_j is within the closest k points to x_i. The K-NN relationship is not symmetrical however, the final graph is usually symmetrised to be undirected. Commonly the outcomes of any graph is represented by a distance matrix A where A_{ij} is the distance between points x_i and x_j.

Once a distance matrix is computed algorithms like principal component analysis or multi-dimensional scaling (MDS) can be used to embed into lower dimensions [156]. MDS utilises pairwise distance from A to from an output that best preserves the distances, usually Euclidean

distance however can be changed based on how A was constructed. Euclidean distance MDS can be shown to be equivalent to PCA [157]. Example of graph geodesic can be seen in one shot embedding algorithms such as Isomap. Isomap being an extension of classical MDS where distances are geodesic, can find more complex non-linear patterns. The calculation of geodesics and the use of a dense matrix adds a degree of computational complexity. Whilst able to preserve global structure, Isomap is unable to navigate complex structures such as holes in any manifold [158]. There are many other algorithms in this field — Local Tangent Space Alignment, Laplacian Eigenmaps and variations, we refer the reader to their respective papers and some surveys [154, 158–160].

Lastly we will be covering two prominent algorithms within this thesis, t-distributed stochastic neighbour embedding(t-SNE) and uniform manifold approximation and embedding(UMAP) [161, 162].

2.13.1 SNE

In order to understand t-distributed stochastic neighbour embedding(t-SNE) we first must explore stochastic neighbour embedding(SNE). SNE assumes a Gaussian distribution between high dimensional points in order to construct relationships between points. A point x_i is considered a neighbour to x_j proportional to the probability $p_{j|i}$ under a Gaussian centred on x_i. The probability of $p_{j|i}$ is:

$$p_{j|i} = \frac{exp(-||x_i - xj||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i}$$

Here the variance of the Gaussian, σ_i, is determined with the help of a user defined parameter perplexity. The choice of σ_i is likely to vary for each high dimensional point x_i as it is likely the density of points surrounding each x_i varies. For dense regions consider less neighbouring points under a Gaussian curve in comparison to sparse points. To achieve varying and suitable σ_i SNE utilises perplexity, a parameter defined by the user. The parameter is akin to a smoothing measure, around each point, for the number of neighbours considered. Perplexity is defined as follows:

$$Perp(P_i) = 2^{H(P_i)}$$

Where $H(P_i)$ is the Shannon entropy:

$$H(P_i) = -\sum_j P_{j|i} log_2 P_{j|i}$$

Likewise in t-SNE the the lower dimensional points y_i and y_j corresponding to the high dimensional x_i and x_j are modelled with the following probabilities.

$$q_{j|i} = \frac{\exp(-||y_i - y_j||^2}{\sum_{k \neq i} \exp(-||y_i - y_k||^2}$$

In order to compare the two distributions $p_{j|i}$ and $q_{j|i}$ from high to low dimensional, Kullback-Leibler is utilised. In essence there is a need to minimise the difference between each distribution to garner a faithful reconstruction of the data. The difference between the sets, higher and lower dimensions, can be modelled as a cost function C

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}}$$

Where P_i is the conditional probability distributions for all x_i and similarly, Qi is all the conditional probability distributions for low dimensional points y_i. It is important to note that the KL divergence is not symmetrical $KL(P||Q) \neq KL(Q||P)$. If two points x_i and x_j are close but y_i and y_j are not the penalty is very high. In contrary, if two points x_i and x_j are far but y_i and y_j are close the penalty is lower in comparison. In summary this non-symmetrical approach encourages the lower dimensional mapping to preserve local structure more than global.

The algorithm optimises the cost function C by means of gradient decent. The gradient of the cost function is as follows.

$$\frac{\partial C}{\partial y_i} = 2 \sum_j \left(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j} \right) (y_i - y_j)$$

We can interpret the gradient as springs between every pair (y_i, y_j). Each spring has direction $y_i - y_j$ and a magnitude $(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})$. If $p_{j|i}$ is large but $q_{j|i}$ is low the difference is positive, so the spring pulls the two points together. Alternatively if similarity is too high in low dimensional space, $q_{j|i} > p_{i|j}$, the spring pushes the points apart. The final algorithm for gradient decent is as follows.

$$Y^{(t)} = Y^{(t-1)} + \eta \frac{\partial C}{\partial Y} + \alpha(t)(Y^{(t-1)} - Y^{(t-2)})$$

where:

- Y^t is the map at iteration t.

- η is the learning rate.
- $\alpha(t)$ is the momentum coefficient at iteration t.

$\alpha(t)(Y^{(t-1)} - Y^{(t-2)}$ adds inertia to the update direction in order to avoid abrupt changes. Additionally in early iteration Gaussian noise is added to the mapping to avoid getting stuck at local minima and over time, t, is reduced, akin to simulated annealing. Parameters such as noise variance, noise decay rate, momentum and step size all had to be carefully selected, usually requiring multiple runs.

2.13.2 t-SNE

t-SNE is an extension of the SNE algorithm, introducing improvements for two problems that exist in SNE. Firstly the "t" in t-SNE represents the use of a student distribution for similarity between two points in the low-dimensional space (y_i, y_j). The heavy tailed distribution alleviates both the over crowding problem and optimisation problems. Secondly t-SNE utilises a symmetrical cost function [163]. The crowding problem is present in any dimensionality reduction algorithm. Consider multiple points in high dimensional space that are equidistant apart, it is impossible to faithfully represent the distances between those points in any lower dimensional space. For example we can consider the Swiss roll dataset, each point can be somewhat faithfully reconstructed to a lower dimensional space given a mapping for each local point. For any complex set of points that are equidistant such as a sphere the ability to map to two-dimensions can't create a faithful global representation.

The authors t-SNE try to combat the overcrowding problem with a heavy tailed distribution in the low dimensional space. The high dimensional space retains probabilities for each point as a Gaussian distribution. The result allows for large distances in high dimensional space to be more effectively mapped in low-dimensional space, unwanted attraction between low-dimensional spaces is reduced. The new definition of $q_{i|j}$, comparative to SNE, is defines as

$$q_{i|j} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

Effectively the Student t-distribution with a single degree of freedom that represents the low-dimensional space allows for infinite mixture of Gaussian distributions with different variables. The resultant KL-divergence gradient is given by the following

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{j|i} - q_{j|i})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1}$$

The resultant gradient has larger repulsive forces for further distances, better separation of clusters and faster convergence. However, distant points still impact the gradient, due to the heavy tail, avoiding the gradient vanishing caused by many small cancelling forces.

2.13.3 UMAP

Uniform Manifold Approximation and Projection(UMAP) is similar to t-SNE where distances are measures via local pairwise distances, in comparison to global relationships such as PCA or MDS [162]. UMAP claims to however produces better global measures in comparison to t-SNE. As for most dimensionality reduction algorithms UMAP approximates neighbours and this process is the most unique element of the algorithms. The initial step being to construct a graph with edges to local neighbourhood points and calculating a weighting between them. The loss function for reduction being to cross-entropy between fuzzy sets, explained shortly. The algorithm is inspired by Riemannian geometry and algebraic topology, assuming the data lies on a manifold and preserve the topological structure.

Given a points $x_i..x_N \in X$ utilise some distance metric d and find the k nearest neighbours. The metric can be defined based on the data and domain, similarly finding and the value of k can be customised. The authors suggest using nearest neighbour decent. Once calculated impose the following

$$p_i = min\{d(x_i, x_{i,j}) | 1 \leq j \leq k, d(x_i, x_{i,j}) > 0\}$$

This ensures that at least one edge has a weighting of 1, enforcing local connectivity in high dimensional space regardless of d. The constraint creates a fuzzy simplicial set being locally connected at x_i. Then the local Riemannian metric is defined by normalising the distances to find σ_i

$$\sum_{j=1}^{k} \exp(-\frac{\max(0, d(x_i, x_{ij}) - p_i}{\sigma_i}) = \log_2(k)$$

σ_i adjusts the Riemannian metric so that each neighbour has a comparable total connection strength. Like t-SNE where each Gaussian is scaled differently for each point x_i so does UMAP. The exponent term is the edge weight from x_i to x_{ij}, subtracting p_i rescales the weightings so that the nearest neighbour weight is 1. Whilst σ_i scales the weighting based on the distance in high dimensional space. Small σ_i results in only very close neighbours getting higher weights

and large σ_i means more neighbours of x_i get significant weights. This can be interpreted as dense regions having higher weights and space regions less so.

This can be then constructed into a graph structure G, with vertices,V and edges E. For a graph $G(V,E)$ the $V = X$ and $E = \{(x_i, x_{ij}) | 1 \leq j \leq k\}$. The graph links all points via a map of intermediary connections between the local neighbourhood of each point. This version of the graph is still non-symmetrical, $x_i -> x_j \neq x_j - x_i$, and is not a global representation. There currently exists only a definition of what each map does however calculating attraction and repulsion between any two points, not just neighbours, is infeasible. To create a unified global representation the algorithms uses symmetric fuzzy union, by combining all local fuzzy sets using

$$B = A + A^T - A \circ A^T$$

Where A is the adjacency matrix of graph G and $\circ$ is the Hadamard product. This creates a final matrix of weights of the higher dimensional space that contains the probability that an edge exists, B_{ij} is the probability that at least one of $x_i \rightarrow x_j$ or $x_j \rightarrow x_i$ exists. The gluing of local metric spaces preserves topology and is justified by fuzzy set theory [162].

Once a the high dimensional representation is constructed projection into low dimensions can be constructed. Similarly to t-SNE, optimisation utilises gradient decent where the cost function is cross entropy. The high dimensional distances are calculated as edge probabilities from B. The low dimensional distance from points y_i to y_j is calculated as follows

$$v_{ij} = \frac{1}{1 + \alpha ||y_i - y_j||^{2b}}$$

where a, b are parameters chosen to match the curve $\frac{1}{1+a(d)^{2b}}$ to user chosen parameters, solving the equation twice for each value of $d = min\_dist$ and $d = spread$. Spread, defaulting to 1, controlling the scale at which distances are measured and minimum distances, usually $\approx$0.1, defining points that should be maximally connected inside of the range. With a measure of distance between both sets the cost function can be defined as

$$C = \sum_{i,j} [B_{ij}\ log\ v_{ij}\ +\ (1 - B_{ij}) log(1 - v_{ij})]$$

Therefore if B_{ij} is large the points move close together and if low points move further apart. In theory this is not calculated for all pairs B_{ij}, N^2 pairs is too expensive. Instead of all pairs the attractive edges are sampled from B_{ij} proportional to their weights and repulsive edges

from negative samples, randomly selected. There is a probability that a negative sample can be randomly chosen as a true neighbours but that probability is fairly small.

In summation UMAP like most algorithms for dimensionality reduction finds the local neighbourhood of each point. Then uniquely creates a fuzzy simplicial set that defines the local mapping of a point, local Riemannian metric. Then lastly for approximating the global structure fuzzy set union is utilised. In comparison to t-SNE even with optimisation UMAP is faster. t-SNE needs all pairwise distances in high dimensional space and has to recompute a Student-t kernel normalisation for every pair in the low dimensional embedding. Umap on the other hand uses fixed pre-calculated parameters a, b and samples for gradient decent , positive and negative, $O(k + n_{neg})$. In finality the complexity of t-SNE with Barnes hut optimisation is $O(N \log N)$ and UMAP is only $O(Nk)$ preprocessing and $O(N)$ gradient decent. That being said the difference in how the data is modelled between each techniques gives utility to both algorithms, neither one is superior when considering a low dimensional space. Where clear cluster separation is more important t-SNE can be utilised, however cluster fragmentation is often frequent. UMAP on the other hand is more practical when a balance between local and global space and is more suitable for larger datasets.

2.14 Summary

Remote sensing platforms and data inherently have many complications to consider. To parse and understand RS data, many initial factors need consideration: the instrument, apparatus, and sensing conditions. Depending on an airborne or spaceborne mission, there is a clear difference between the resolutions. The flight direction and configuration of the apparatus play a vital role, whether the sensor is facing directly down, along-track, angled, cross-track, or in a multi-layout configuration. Active and passive sensors also provide a distinct difference in resolutions with the sensed portion or portions of the electromagnetic spectral being crucial for understanding the earths surface. In addition to all those initial complications, the earth's surface introduces more variability. Namely the extreme quantity of different features and each individual feature's ability to have different orientations, sizes and layouts which all contribute to the unyielding complexity of RS data. Therefore, ideal datasets should have high intra-class and interclass variations, leading to larger and larger datasets. Datasets are utilised for three overarching aims: anomaly detection, background characterisation and target recognition, where labelled data is crucial for training and testing. Where RS data is one of the most complex data sources, labelling requires expert analysis. Accurate labelling, therefore, incurs

a huge time and financial cost.

For spectral imaging, labelling can be defined as the physical process of comparing different wavelengths altered by the material they interact with. Machine learning approaches consider edges, regions, thresholds, or clustering differences between sensed wavelengths. Labelling tools can provide a method for preprocessing data for downstream model ingestion or aim to further label data. Each dataset requires a user to have a predetermined set of goals and features for extraction, leading to the creation of datasets with limited labelled categories or classifications.

Finally the background concludes with common techniques utilised within the thesis. Explained are autoencoder, relevant to early technical works which then evolves to graph neural networks for their ability to encode relational information. Underpinning the entire thesis is the use of manifold learning techniques to create two-dimensional visualisations which can be seen throughout all works.

Chapter 3

Literature Review

Contents

Remote sensing now produces vast quantities of imagery however, creating high-quality labelled datasets remains a critical bottleneck for downstream applications. Preprocessing and annotation pipelines are still highly manual and application-specific, complicated by domain characteristics such as multi-sensor variation, temporal resolution, multi-label scenes and heterogeneous class boundaries. The literature includes tools for preprocessing, systems that leverage external sources and automatic labelling for dataset construction.

To reduce human effort, methods from information retrieval and content-based image retrieval are used to retrieve visually similar samples from extensive archives. However, they inherit limitations related to bias, class imbalance, and semantic uncertainty. Iterative annotation approaches, such as relevance feedback and active learning, further reduce the need for expert involvement. While generative and metric learning address feature space modelling. Finally, visualisation remains a significant challenge as RS visual applications are map-centric and image-focused, which are not conducive to mass labelling and feature comparison. Together, these themes highlight a fragmented landscape in which no single tool enables scalable and exploratory dataset construction.

3.1 Labelling Tools

3.1.1 Pre-Processing

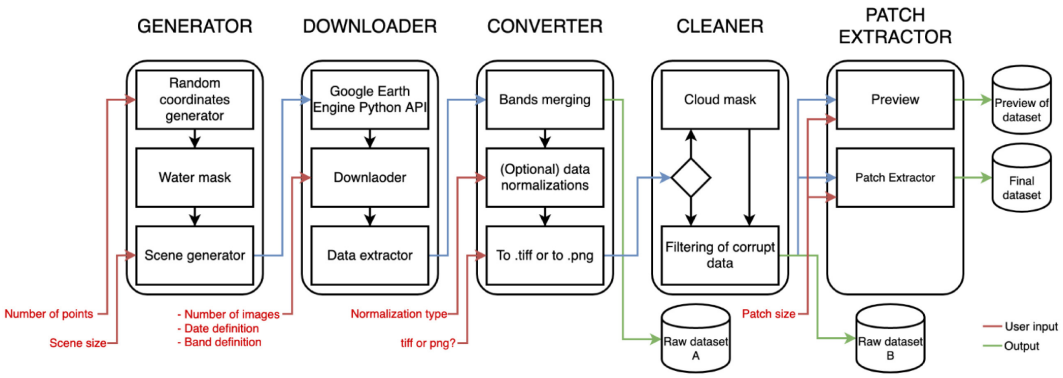


Figure 3.1: Example of preprocessing for dataset creation from [7].

The most time-consuming and "least enjoyable" part of dataset creation is the cleaning and organisation of data [7]. In order to alleviate the time-consuming processes which are involved in cleaning data, Sebastianelli created a tool for automation, shown in Figure 3.1 [7]. The images are sourced from Google Earth Engine, guided by user-entered longitude and latitude

points or randomised. Location-based selection of images assumes that the user has a particular site they wish to study. Randomised selection of images is rare in remote sensing for one reason: most models are geographically locked. Creating geo-generalisable models is currently a major challenge within remote sensing. With no downstream labelling incorporated within the tool, this tool is more so a precursor to labelling. Alternatively, the tool could be used to generate test datasets to verify models with user verification if no labels are present, both avenues requiring labelling a costly task to which this tool offers no support.

Second to image selection, locational or randomised, is adjusting the water content. Water or cloud content is normally one of the first filtered elements from any dataset. As the majority of the Earth's surface is comprised of water, applications can filter to include or exclude, depending on the use case, for example, land use or offshore installations (oil rigs, windfarms, etc.). Water masks can be supplied from certain acquisition platforms, such as the Sentinel program, where water masks are an inherent preprocessing step, or manually processed from water indexes. Water indices such as the normalised difference water index(NDWI) [164] or Sentinel's own Sentinel water index(SWI) [165] return a value based on the differences between multiple bands, see section 2.6. While simple index-based thresholding remains dominant, it is highly sensitive to location, seasonality, and sensor modality, limiting generalisation across geographic domains. Further challenges to indexes in water detection are factors such as topography, snow, ice, clouds, and shadows which cause noise and compromise the effectiveness of this approach [166]. Reliance on pre-computed platform water masks or models can lead to systematic error propagation into the dataset.

Where this tool currently uses water masks for simply detecting the presence of any water, it could be expanded to detect differing water features, lakes, rivers, rivers with sediment, calm waters, ice and more. Each variation hold a unique insight and applicability to differing datasets which cannot be captured using a singular index. The dataset creation tool could be expanded to contain multiple different indexes or multiple joint indices, for water there are many [166]. Machine learning and AI models have been utilised to combat water detection within some of the aforementioned adverse conditions to varying degrees of success which could also be incorporated into the tool [3]. AI models also come with limitations on their generalisability, dependent on their training set. This process could also be extended to other indexes such as chlorophyll, snow or marine cyanobacteria and more, see section 2.6. Each index has its own applicability given the imaging conditions and geolocation, which could be encoded into the dataset creation tool to either automate or allow for guided user selection.

After location selection, the user is able to select the number of images, bands and the format to export from the raw ingestion. Band selection for products from hyper spectral imaging is crucial to reduce the ingested data to only the more important information [167]. Band selection does not only reduce data volume; it directly affects class discrimination and therefore the quality of downstream learning. Normalisation is optional in the form of min-max, standardisation or max normalisation, to assist any post processing for machine learning and AI applications. High reflectance materials, such as metal, can also produce anomalies producing higher reflectance values above the maximum range. Normalisation confines anomalies to within the acceptable range of values. Alternatively, an AI method could also be utilised for de-noising, to not only confine pixel values to acceptable ranges but to combat any "bleed" that occurs from complex noise to surrounding pixels [168].

In order to extract cloudless data, the tool can search through different temporal resolutions via the provided mask by Google Earth Engine products to refine the returned results. The cleaner stage ,in addition to cloud removal, removes any partial images and corrupted images that may exist. Partial images can be found where the swath of the satellite is not covering the entirety of a UTM zone in which Google Earth Engine images are parsed into, see section 2.3 for more details on swath. Finally, as most models are tested on a train and validation set, the images are organised into sets by the dissimilarity, utilising the cumulative histograms of the two sets. The tool is limited to non-semantic splits of the dataset as no labelling or feature extraction is introduced to incorporate any semantics. A simple extension could look to compare both textural and spectral histograms to split data into more informative training and test sets.

This approach shows the backbone of filtering through data for specific data when a target location is known. The format, size and integrity of data are important at this stage. Whilst this approach disregards data which have been flagged by clouds by utilising the mask provided by Google Earth Engine, there are other methods to detect or remove clouds. Cloud detection has been done before utilising physics-based approaches such as difference indexes or more complicated systems that also detect shadows from clouds [169, 170]. Example recent deep learning approaches consider U-nets with spatial attention mechanisms or VGG16 based classifiers [171, 172]. Sun et al utilise a combined approach of physics-based and NN to classify different, thin or thick, cloud cover types such as cirrus or cumulonimbus [173]. Where the physical modelling created a spectral library for the model to utilise for generalisability, other methods utilising generative or AE models aim to remove cloud clover entirely [174–178].

Some methods utilise spatial attention blocks [176, 178]. Whilst some others utilise larger temporal resolutions [178].

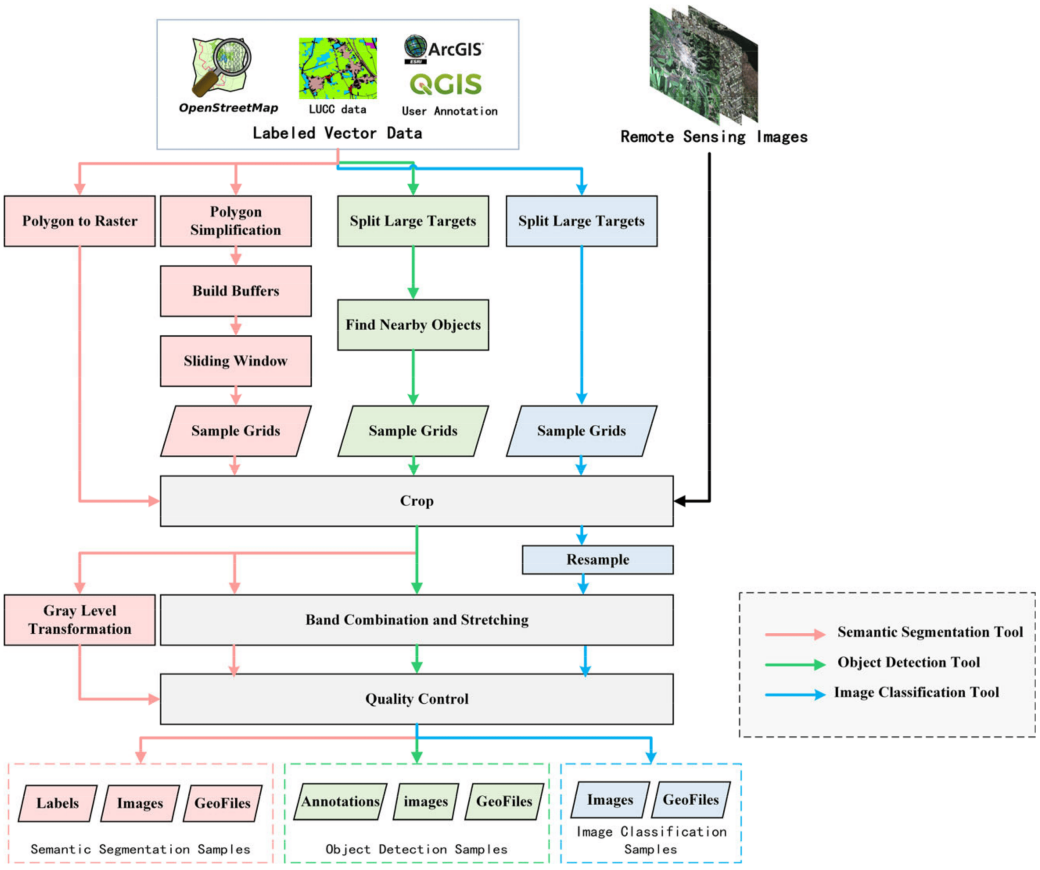


Figure 3.2: Pipeline of LabelRS system [8].

3.1.2 Data Annotation

Building on basic satellite image preprocessing, there is an introduction to labelling. Li et al. created a solution for a labelling system for dataset creation [8]. Notably, the approach splits the pipeline into three avenues: semantic segmentation, object detection and classification (Figure 3.2). These three avenues cover the main dataset configurations needed for most downstream ML and AI tasks (see Section 2.4). The application utilises existing frameworks such as OpenStreetMap (OSM), land use and land cover change (LUCC) data and ArcGIS. The foundation interface is built on ArcGIS, where users can label and add annotations to remote sensing images or derive labels from OSM and LUCC data.

OpenStreetMap uses crowd-sourced labelling to address the annotation problem, but this approach has varying suitability depending on the application. A study found that higher population density areas labelled via crowd-sourcing are more commonly updated and have higher accuracy, whereas certain land use/land cover (LULC) classes, such as land, have weaker accuracy [179]. Whilst open-source data is updated regularly, it is highly dependent on geographic and socio-economic biases, poor or rural areas are rarely updated. There also exists variable annotation consistency when considering the many different user annotations applied, and maintaining an enforced labelling standard is infeasible at scale.

Most open source data is kept up-to-date however, this creates a temporal resolution mismatch between the dates the labels were applied compared to a users needs. For the aforementioned reasons, open-source data can hardly be considered ground truth and is highly limited by the predefined categories imposed by users. LUCC faces similar issues where the labels may drastically change between the label generation and the acquisition of a users image. The paper does not discuss how the quality of aggregated labels is verified, nor how conflicting annotations are resolved. Despite the goal of automating dataset creation, significant manual intervention is still required, which may limit scalability and reproducibility.

Within LabelRS, all user and pre-existing labels are utilised to create semantic segmentation. The paper defines a methodology to split and merge samples relative to how dense or sparse their distribution is. Optimisation is aimed at splitting densely distributed targets into independent objects, where the user has coarsely applied a rectangular bounding box over a region, and alternatively finding sparse objects scattered within a region. Both methods utilise a sliding window with a radius buffer. This approach to semantic segmentation is consistent with literature, however, there is no consideration for balancing datasets between classes or modelling variation within classes.

Object detection is conducted around a radius of pre-labelled samples using PASCAL VOC, YOLO and KITTI. The applicability of these object detection models is not reviewed or tested to determine whether they find all instances of objects within a scene. Finally, image classification uses a windowed approach to generate samples for classification. Whilst each of the three label formats is appropriate for major remote sensing use cases, there is no inclusion of sample variation, train/test splitting of datasets, or quality control for automatic bounding-box detection.

Lastly, we show an example of how a tool can be utilised to label data from a large region automatically. Li et al. introduce a pipeline, see figure 3.3, for automatically labelling

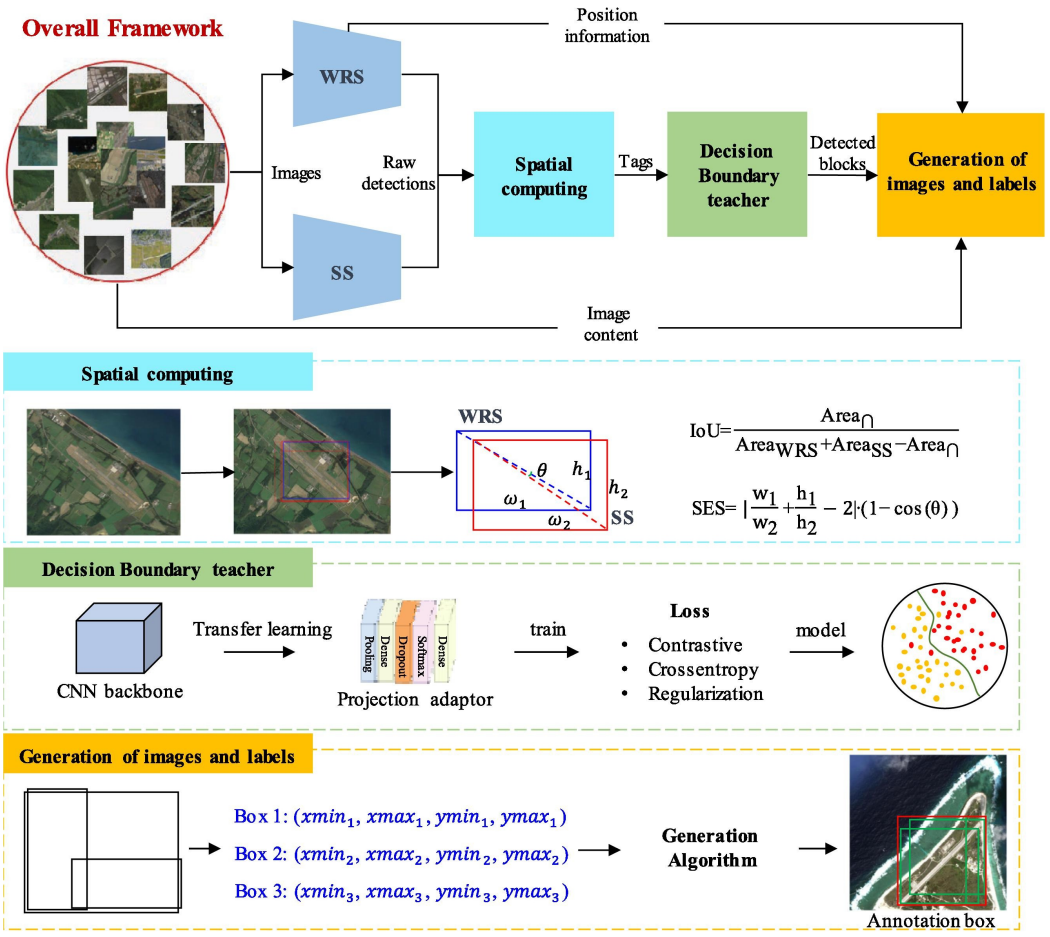


Figure 3.3: Pipeline of an automatic labelling framework [9].

samples across larger geographical regions. They utilised a pre-existing dataset DIOR, which includes 20 classes and 23,464 images annotated with a bounding box. The data was used to train a dual student model that initially identifies objects of interest. The pipeline does not remove the need for labelled data, it transfers their dependence to existing datasets and inherits their class distribution and bias. This model bias includes the inability to transfer to arbitrary geographical locations, the apparatus or temporal variations. A spatial computing block calculates the degree of similarity of graphical similarity between bounding boxes of the same class for finding high-confidence labels. The confidence labels are not conveyed to the user, it is in these uncertain samples that variation and richness of samples are often found. Similarity of features within a bounding box is calculated using the intersection over union (IOU) and the ShapE similarity (SES). Where the complexity of background objects can confound the

similarity between two regions, an encoder with a ResNet50 backbone is used to estimate a decision boundary. The model is optimised to replicate the dataset's annotation style, not necessarily the ground-truth geometry. Finally, for image generation and labelling, an advanced YOLOv7 network is utilised. The pipeline overall fails to include any handling of cloud or shadow occlusion or any temporal sensing variations.

Each of the three applications discussed shows a gradual evolution of functionality towards the classification and labelling of data without human supervision. The first demonstrates the steps needed to preprocess and build a dataset without labelling. The second application, LabelRS, provides multiple different formats (semantic, object or image classification). The use of open-source data, however, can introduce errors, especially in rural areas. Open source data comes under more scrutiny when even experts do not come to a consensus when labelling data [16]. In addition, open-sourced data is limited by the pre-defined classification categories. The last application focused on a broad area classification however, it is bound to pre-defined classes utilised within the training process.

3.2 Image Retrieval

Content-based image retrieval (CBIR) attempts to find and organise images based on the content or semantics within a dataset of images. The retrieval process is based on the needs of the user and an initial selection of a query image. CBIR methods are used at the instance level where the image contains a single object or at the class level where multiple objects from different classes can be present. A CBIR system comprises three main parts: feature representation, feature hashing and the final retrieval stage.

Early work extracted feature representations from intermediate stages of popular classification models such as ResNet and VGG. These models allowed either off-the-shelf feature extraction or fine-tuning of the model as the backbone for feature embedding. However, the use of models optimised for classification introduced the domain shift problem, particularly for remote sensing (RS). The activation maps learned for classification do not necessarily transfer to retrieval, where the goal is similarity rather than categorical separation. Feature extraction from convolutional maps was therefore refined and aggregated using multiple methods. Babenko introduced sum-pooling convolutional features (SPoC) with Gaussian centring to aggregate features [180]. Earlier methods such as bag-of-words (BoW) were also used [181,182], but were soon replaced by VLAD for more compact and discriminative representation [183].

VLAD inspired later work that implemented aggregation as a fully connected layer [184], thereby reducing computational cost for increased training time.

Activation maps for each class targeted by the loss function, while suitable for classification, can be refined for retrieval. Class activation maps (CAM) added relational weighting across class-specific maps [185], combining the benefits of CRoW and R-MAC [186, 187]. Other works adopted parametric attention-based approaches to extract saliency maps [188]. Despite these improvements, all approaches remained limited by their dependence on training data and backbone CNNs. These models can struggle to encode the multi-scale objects, spectral variations and mixed semantics that are characteristic of RS imagery.

A survey conducted by Dubey found more training schemes than classification outputs, including autoencoders (AEs), Siamese or triplet networks, and generative or recurrent models. Basic latent-space features extracted using AEs have been used for CBIR [189, 190]. More recent approaches used denoising AEs in the medical domain [191]. Singh et al. utilised triplet loss combined with AEs for medical imaging [192]. These approaches often struggle to encode spatial detail without additional supervision or attention mechanisms.

Siamese and triplet-loss training schemes have become popular as they exploit the distance between features within the embedding space. Siamese networks operate on pairs of samples, whereas triplet networks use three samples to push or pull embeddings closer or further apart. The underlying principle is that each class should occupy a well-defined region in the latent output space. Examples include medical imaging [193, 194], cross-source retrieval using a Siamese transformer [195], and unsupervised retrieval with transfer learning [196]. However, these approaches rely on careful negative sample mining and can be unstable for multi-label scenes. RS imagery often contains multiple semantic categories within a single image, and defining similarity is difficult when scale and context can be drastically different between images.

Information retrieval (IR) methods for RS have recently seen rapid growth [197]. Whilst not fully automated, an initial image or a small set of samples can be provided to extract similar features from a large archive. IR methods, therefore, address both the feature extraction process and how features are stored. With the vast volume of RS data, search speed and memory are critical considerations. Most recent methods use CNN feature extractors [198–203]. Some works utilise spatial or spatial-channel attention for improved feature localisation [199–201]. Others model relationships between features using graph neural networks [198]. When retrieving images from the projected feature space, similarity between samples relates to the

probability of false retrieval. Triplet and contrastive losses are often applied to reduce this probability and enforce separability [200, 203]. Graph-based approaches such as region adjacency graphs (RAGs) segment the image into superpixels and link neighbouring regions [204]. These approaches can model structural context, which is a major challenge in RS.

Whilst IR systems are highly effective, they do not address several key requirements for automated dataset construction. Firstly, the requirement for an initial query image assumes that the user already possesses relevant examples. For multi-class dataset construction, many queries may be required across features such as roads, buildings and vegetation. Secondly, a single query tends to retrieve the most common samples with minimal variation. For example, querying an urban region repeatedly will often return visually similar scenes, limiting inter- and intra-class variation, which is crucial for robust datasets [2]. Identifying slightly dissimilar examples within one class requires thresholds to be set, but it is unclear whether the tenth returned result or the thousandth retains the same semantic class.

Generalising the feature representation is another challenge. CNN-based features struggle with multi-scale problems, failing to capture both local detail and global context simultaneously. RS images can contain extremely small rooftops alongside large forests, a single embedding may not effectively represent all features, large and small, within a single image. Semantic understanding is difficult due to complex objects and multi-label scenarios. The feature hashing and retrieval stages must also handle searching large-scale database operations.

3.3 Iterative Annotation

Image retrieval approaches can be improved through the integration of user feedback during the labelling process. Relevance feedback can be an extension of image retrieval systems to address the semantic gap between the user's intent and the low-level visual features returned by a query. Works in this area aim to extract information from a given query to improve future queries. However, relevance feedback still assumes that user interaction reliably reflects semantic similarity and does not fully remove the semantic gap, particularly for multi-label or heterogeneous RS scenes.

Information can be explicitly extracted by allowing the user to suggest positive and negative samples, or implicitly through time spent viewing, clicks or scrolling. One limitation of such mechanisms is that they may introduce user bias or reinforce skewed representations when the initial query is unbalanced. Once user intent is modelled, we can apply methods such as triplet loss to retrain the model using hard and negative samples [205]. Similarly, Henkel et

al. utilise pairwise samples and contrastive learning from a labelled dataset with user guidance for uncertain samples [206]. Updating an entire model improves classifications for the target domain; however, full model updates may be computationally expensive and difficult to scale to large RS archives. For a more general approach, only the final layers are updated, as in Shabbir et al., who fine-tune only the final layer of ResNet50 [207].

Relevance feedback has still not been widely explored in remote sensing, as found by Zhou et al. [208], reflecting a wider trend that most approaches are evaluated on natural-image datasets rather than multi-sensor or multi-season RS scenarios. In computer vision it remains prominent. Nara et al. utilise CLIP with relevance feedback for each user's preferences [209]. CLIP is used as an encoder to hash images into vectors, and the updated query is handled by a 1-NN classifier requiring no additional training. CLIP-style systems, however, inherit biases from natural imagery and may not capture spectral or topographic cues unique to RS. Other methods exist for updating image relevance from multi-modal models, retraining and differing relevance metrics [210–213].

Active learning, in contrast, is well established in remote sensing and computer vision. Whereas relevance systems refine user intent, active learning trains a model with minimal labelled samples by querying the user (oracle). Active learning systems proactively prompt the user to label informative samples. Models trained in this way are updated when needed and require only small labelled subsets, reducing expert analysis [214]. A limitation is that AL assumes access to reliable annotators and does not inherently guarantee class balance or variation, which are important for RS datasets.

Pool-based sampling selects uncertain samples from an unlabelled set using auxiliary networks, prediction fluctuations or gradient norms [215–218]. Möllenbrok et al. combine these methods in a multi-label classification model [216]. Siamese networks have been used for hyperspectral segmentation, where one model is contrastive and the other classification [219]. In practice, uncertainty metrics can oversample visually similar or redundant samples, and informative sample selection remains an open problem.

Another strategy uses generative models such as AE, VAE and GAN. For small datasets, synthetic samples are invaluable, as shown in other fields [220, 221]. In remote sensing generative active learning is rare. RS generative models are mostly used for fusion, super-resolution, denoising and augmentation [11, 222–224]. Generative samples must preserve spectral and radiometric consistency, otherwise artefacts compromise analysis. Consequently, generative models in RS have found their strength in restoration and enhancement. Early examples such

as MetaEarth show potential for synthetic EO data [225]. Hybrid approaches also exist, where generative models support pool-based AL [226].

Active learning has also been adapted to data streams. Stream-based sampling selects the most informative incoming samples and is used in real-time systems. Examples include LiDAR point-cloud applications where samples are selected using Shannon entropy [227], and crowd-sourced labelling near the decision boundary [228]. Stream-based approaches depend on continuous data availability and may accumulate errors during long deployments.

In summary, iterative annotation refines models at the pre- or post-deployment stage by selecting informative samples, based on user interaction or uncertainty. Samples may be generated or found outside the initial dataset, although generative images in RS are limited by reflectance physics. Updating models is commonly performed via contrastive learning, selecting hard positives or negatives. The resulting feature space is more suitable for separating predetermined classes, but generalisation to other domains becomes difficult. Remote sensing images are complex and targeted training comes at the cost of generalisation, which is important due to the many applications in the field.

3.4 Datasets

Remote-sensing datasets are the curation of images from raw satellite feeds for computer vision approaches in Earth observation. While section 2.5 outlined the sensing platforms, spatial resolutions and modalities available this literature section will discuss how these raw streams are organised, curated and benchmarked for machine learning applications. As a result, dataset choice is now recognised as a methodological decision rather than a passive constraint, influencing feature separability, model generalisation and labelling strategy.

Most datasets are curated with a specific downstream task in mind, classification, semantic segmentation, object detection or change detection, as discussed in section 2.4. Most research datasets are often deliberately geographically constrained to simplify annotation by ensuring manageable class balance or highlighting a particular land-use pattern. This contrasts with operational satellite archives, which prioritise global coverage and temporal frequency rather than their machine learning suitability. Consequently, there is a dichotomy between scientific representativeness and label efficiency.

As discussed in section 2.5, inter-class and intra-class variability significantly influences the robustness of learned models. The literature reinforces the variability observation, small homogenous datasets frequently lead to spatial over-fitting whilst large-scale datasets, those

spanning multiple geographical locations, seasons or sensors, enable models to generalise beyond the region of acquisition [229, 230]. However, increasing the dataset diversity also raises challenges around class imbalance, domain shift and harmonisation of spectral statistics. These problems can be especially prevalent when merging imagery from different satellites [231].

A recurring observation in recent research is the scarcity of high-quality remote sensing labels, despite the extensive archives of imagery available. Annotation remains a time-consuming and arduous task, which often requires application-specific expertise, leading to fragmented and unevenly distributed labelling efforts. This limitation has motivated the development of weakly supervised, self-supervised and unsupervised labelling strategies. Many of these methods explicitly aim to reduce dependence on manual annotation. Annotations are not only evaluated on accuracy but also on their granularity, consistency and semantic stability across geographical regions and temporal resolutions.

While section 2.5 categorises satellites by spatial resolution, revisit time, and sensor type, this section emphasises the use cases. Multi-spectral data imagery is commonly utilised for vegetation monitoring and land-cover mapping. As the bands are very limited compared to hyperspectral data, multispectral is more generalised in its use cases. On the contrary, hyperspectral data is utilised for fine material discrimination. Where hyper-spectral data contains hundreds of bands some form of dimensionality reduction is required. SAR is most commonly used to map areas that are under cloud cover, due to its ability to penetrate through cloud coverage. Likewise, LiDAR is primarily used to map multi-layered areas, such as forest canopies and ground foliage. The sensor modality affects both the difficulty of dataset construction and the nature of the learning problem. In an unsupervised setting, the spectral structure, noise and acquisition conditions act as implicit supervisory signals.

A key theme emerging from recent papers and datasets is the lack of universal benchmarks within remote sensing. Unlike general computer vision, where ImageNet or COCO standardised model comparison, remote sensing research relies on datasets of various resolutions, class definitions and geographic scope. As acknowledged in section 2.5, many commonly used datasets originate from specific regions. There is an increasing need for future datasets to prioritise geographical coverage, sampling strategies and multi-class annotations.

Im summary, this review of datasets highlights that dataset selection is not merely a practical constraint but a methodological detriment to model behaviour. For fast labelling systems, especially unsupervised systems, the data must provide sufficient spectral, spatial, and temporal diversity to enable meaningful pattern discovery.

3.5 Visualisation

The visualisation of complex multidimensional data on a two-dimensional display is a challenge across the remote sensing and computer science domains.

3.5.1 Remote Sensing

Maps and cartography are the traditional means of conveying remote-sensing information. Historically, as illustrated by the UKHO example, arranging images into their geographic locations has been the most effective way to contextualise data. Maps allow users to identify an area of interest and retrieve imagery for interpretation. This also enables the concept of tiling—dividing the Earth into rectangular units. An example is the MGRS tiling scheme. Applications can render the Earth as a global three-dimensional object, but any meaningful inspection requires zooming until the view collapses to a two-dimensional plane.

This paradigm embeds several constraints. A user must already know the location they wish to explore. Navigation becomes manual, relying on pan–zoom interactions, which scale poorly when searching over large areas or multiple scenes. Spatial proximity is assumed to imply semantic similarity, which does not always hold in RS (e.g., two adjacent tiles may contain different land cover or sensor conditions).

Visualising remote-sensing imagery is demonstrated in SNAP (Sentinel Application Platform). SNAP allows users to load products, inspect metadata, view composite bands, batch-process imagery and generate spectral indices. It also includes maps for footprint visualisation and layer managers for toggling land-cover masks, vector annotations and composites. SNAP reflects the dominant interaction approach in RS: panning, zooming, and layer-based interaction, supplemented by preprocessing operations such as atmospheric correction or radiometric calibration.

However, most tools in this domain share fundamental limitations. Interactions are primarily on a single scene or a small set of scenes, requiring users to scan through large areas manually. The focus is on image navigation rather than data exploration. Visual comparisons are often restricted to geographically localised neighbourhoods, which scale poorly to large datasets, especially those that are geographically diverse. RS visualisation environments are designed for single-image analysis rather than large-scale dataset construction.

3.5.2 Computing perspective of visualisation and dimensionality reduction

The previous section introduced the basics of exploring and visualising RS images. For computer scientists this definition can be expanded. Classical foundations include human–computer interaction [232, 233], effective visual communication and design theory [234], and more recently annotation and labelling visualisation [235]. Here, we focus on the visualisation and interactivity of two-dimensional projections of points. This arises from a gap in research: the inability to effectively communicate relationships between multiple samples within a dataset.

A dataset, in order to be efficient for downstream AI applications, needs inter- and intra-class variation. Current methodologies such as recommendation and retrieval systems do not convey variation sufficiently. Tools discussed in Section 3.1 only find instances of classes. Users may infer variation from a single sample (e.g., by ranking K-nearest neighbours), but this neglects variation between multiple samples simultaneously. Pipelines typically embed images in vector spaces and sample or modify those spaces. The problem is therefore not entirely building high-dimensional spaces, but visualising relationships within them.

Human perception and the two-dimensional display limit direct exploration of high-dimensional space. To perceive relationships in two dimensions we turn to manifold learning (see Section 2.13). Manifold learning extracts a nonlinear low-dimensional manifold from the high-dimensional space and approximates it into a low-dimensional plane. Even a well-behaved Riemannian manifold will stretch or tear when flattened from a manifold learning algorithm when projected into a low-dimensional space, hence all methods are an approximation. Usually, for common methods such as t-SNE or UMAP, local neighbourhood structure is captured more accurately than global relationships.

Whilst not yet prevalent in remote sensing, there are many examples within computer vision. The most common non-interactive use is to verify embedding spaces [146, 219, 236]. Visualising embedding spaces provides, at a cursory view, the relationship between points, if colour-coded as many are in a classification example. A deeper understanding of these embeddings can be achieved by analysing the content of each point and enabling interactive exploration in the visualisation tool. For interactive exploration, Sacha et al. surveyed 377 papers [237] and identified scenarios including annotation and labelling. Examples include starSPIRE, where text documents are visualised for iterative refinement [238, 239]. Interaction typically consists of moving, selecting, annotating or drawing boundaries between points, with labels refining the model in subsequent iterations.

Interaction has also been used to guide t-SNE during computation. The projection of millions of points is expensive, so Pezzotti et al. introduced approximations using fast KNN computations to reduce runtime [240]. The user can dictate approximation levels or specify subsets of points, steering the algorithm. Their application includes methods for inserting or deleting points.

In summary, many computer science approaches utilise smaller image patches to build searchable pools of imagery. Retrieval or recommender systems cover large areas, but smaller image tiles offer better coverage than map-based navigation. Lists of retrieved results hinder understanding of dataset-wide variation. Manifold learning addresses this by projecting image relationships as points in two dimensions. The interaction involves selecting, annotating, moving or boundary-drawing. Moving a point does not translate to a physical coordinate in the original space, but creates new local relationships for updating the pipeline. Selecting points both reveals their content and guides optimisation. Dimensionality-reduction techniques can be time-costly, so techniques exist to guide the algorithm based on user intent [240].

3.6 Conclusion

Across remote sensing and computer vision, significant progress has been made in automating annotation, retrieval and exploration of imagery. Tools now support preprocessing, segmentation, automatic sample generation, retrieval-based exploration and iterative annotation. Yet major limitations persist: reliance on external labels, bias and domain shift, poor modelling of variation, difficulty handling multi-label scenes, and limited support for dataset-level visualisation.

Most current systems focus on isolated stages of the pipeline rather than providing an integrated solution. Retrieval tools produce similar samples, but not relationships. Iterative annotation trades generalisation for task-specific refinement, and visualisation approaches struggle with scale and semantic clarity. Thus, despite advances, the core challenge remains unresolved: the ability to efficiently explore, curate and label large unstructured remote-sensing archives in a scalable and semantically meaningful way.

Chapter 4

Problem Statement

Remote sensing generates massive volumes of imagery from diverse technologies (satellites, UAVs, aircraft) and apparatus (lidar, radar, multispectral, hyperspectral). Datasets are essential for applications such as land cover mapping, disaster monitoring, and environmental analysis. The domains utilising remote sensing technology are expanding faster than the curation of applicable datasets. The bottleneck in remote sensing lies in labelling, both for existing and potential new datasets.

Manual annotation is costly, time-consuming and inconsistent, especially for large-scale satellite imagery where spatial resolution and class ambiguity introduce uncertainty. As the area of studies expands from towns, countries to continents, feature variability increases significantly across regions, climates and sensing conditions. Therefore, most existing datasets contain limited geographical coverage and are curated for narrow tasks or alternatively, large-scale studies contain generic and inaccurate labelling. The applicability of transferring labelled data between apparatuses is infeasible due to spatial resolution differences and evolving sensing conditions. The same can be said for data acquired from an identical platform and apparatus; different temporal resolutions require unique labels, given the same geographical region.

Current research for alleviating this problem is broad. Classification models can be trained on limited samples of data and applied to unseen samples to find more instances of the same class. Classification models have been seen to be utilised successfully, they still require an initial pool of data, requiring costly expert time and sacrificing generalisability. Other solutions are image retrieval or recommender systems, which don't necessarily require any initial labelling samples. With a generalised model of embedding images, retrieval systems can be queried with an initial sample to retrieve similar results. The mentioned methods and models

can be optimised iteratively, most commonly through user feedback and learning user intent. Accurate iterative optimisation require ground truth labelling, for remote sensing this is almost infeasible due to inherent uncertainty. As remote sensing datasets continue to grow, these approaches do not scale proportionally to the volume and diversity of imagery produced.

Each downstream application within ML or AI approaches require a dataset to have intra- and inter-class variance which is not explicitly modelled within the current solutions in literature. For all classification, retrieval and active learning models, an initial dataset containing variance rich samples across regions and temporal resolutions would alleviate the need for increased labelling. Therefore the key challenge is modelling intra- and inter-class variance to build robust initial datasets. Whilst manifold learning offers some promise in modelling variance, no scalable framework exists that explicitly captures variance across large satellite datasets.

An ideal solution would provide a compact, generalisable image encodings that represent all features. Compact encodings reduce the computation and memory cost of any downstream analysis, considering the scale of RS imagery. Whereas generalisability addresses the need for remote sensing images across multiple unique domains. The solution should also be malleable to the three main labelling paradigms, scene, object and pixel. Additionally any methodologies must be applicable to the different sensing modalities(e.g. SAR, spectral and Lidar). Each modality could be considered individually or ideally combined, however, this may be infeasible due to the uniqueness of each apparatus, especially three-dimensional point clouds produced by Lidar.

This thesis addresses these challenges by developing a scalable labelling tool that enables efficient exploration and comparison of large-scale satellite imagery, focusing on variance modelling, similarity search, and dataset overview. The aims are formalised in the following research questions:

- **RQ1**: How can unsupervised and weakly supervised methods be used to construct scalable, variance-aware representations of satellite imagery suitable for large-scale labelling?
- **RQ2**: What unified encoding framework can compactly represent scene-, region-, and pixel-level information while remaining generalisable across different types of remote sensing data?

- **RQ3**: How can similarity-based retrieval, manifold exploration, and neighbourhood-aware sampling reduce annotation effort and support efficient dataset curation?
- **RQ4**: What novel interaction paradigms and tooling can reduce expert workload while enabling production of high-quality labelled datasets suitable for downstream remote sensing applications?

Chapter 5

Unsupervised scene sample extraction

The works in this section were published and accepted as Tulsi Patel, Mark W. Jones and Thomas Redfern, Manifold Explorer: Satellite Image Labelling and Clustering Tool Using Deep Convolutional Autoencoders, Algorithms 16(10):469, 2023 [29]. A video was also published alongside the works, aiding in understanding the visualisations and interactions. The introduction has been completely revised, only fragments of the old version remain.

Within previous sections of this thesis, we have defined the major problems found in labelling tools and explored how to ingest remotely sensed data. Existing labelling applications fail to capture or visualise the variance in data between samples. With most applications taking a non-generalised approach utilising pre-labelled data, at a cost to both time and expertise.

As the first technical work, this section looks to build a foundation for the labelling tool. This work is focused on processing images at the scene-level, as opposed to object or pixel classification. The motivation for scene level labelling is simply to reduce the complexity of the problem set and demonstrate that the foundational pipeline can provide an effective solution before diving into more granular features in future chapters.

This chapter contributes a novel unsupervised labelling framework that combines autoencoder-based feature learning with manifold visualisation (t-SNE and UMAP) to enable human annotation of scene-level remote-sensing imagery without pre-labelled data. We examine the utility of models to encode images and a visualisation framework for labelling. The assumption being that remotely sensed data, like most datasets, has samples that lie close to a manifold in high-dimensional space. Remotely sensed images, despite their high spectral dimensionality, tend to occupy only a small subspace of the input space. Spectral reflectance is constrained by surface materials and sensor physics, implying that meaningful image vari-

ation lies on a lower-dimensional manifold. However, we cannot assume that the complexity of the manifold is suitable for generalising all pixels, remotely sensed images contain far too many factors that cause variation between pixels. Most works do not generalise for all pixels; they utilise labelled images to find manifolds or the boundaries between subsets of the data. With our approach looking to alleviate labelling we must consider all possible data. As stated before, this section looks to find scene-level similarities, reducing the complexity by removing individual pixel variation.

For the labelling application we have also found previously in the literature review, chapter 3, that manifold learning is the best suited methodology for exploring and visualising high-dimensional spaces and therefore this work looks to deploy such algorithms. There are two main inputs to such an algorithm. The first input is the high-dimensional vectors, and the second is the function for which distances are measured between these vectors. Thus in order to utilise manifold learning we must consider how to encode our images into meaningful high-dimensional vectors. We must also consider the size of the vectors for computational efficiency. The effectiveness of the pipeline is both in the effectiveness of the image encoding and its ability to form a manifold in high-dimensional space for meaningful reduction. If both criteria are met, then we would expect to see a visualisation that allows for exploring similar images and their relationships to all other images.

We utilise an AE architecture to represent our features, focusing on defining how the user can navigate the complex latent space created by multiple convolutions. We propose a solution that uses embedding the higher-dimensional space into two dimensions, using t-SNE and UMAP, to visualise inter- and intraclass variation between images accurately. Conducting a user study to prove that the ease with which a user can navigate a dataset for labelling is improved. This approach has been considered concerning time-series data [241, 242], and time-series clustering [243]. Whilst the previous approaches considered biological time series data, diving birds, this work is focused on remote sensing images that contain no temporal patterns. The following background section reviews existing methods for feature extraction and discusses why autoencoder-based architectures are well-suited to the task.

5.1 Background

5.1.1 Unsupervised feature encoding of images

Feature encoding of remote sensing images is based on finding either textures or spectral features of interest within an image. The most fundamental and historical method of finding samples of interest from within images is hand-crafted features, see section 2.7.1. Beyond generalised descriptors, hand-crafted features can be optimised to include specific subtleties within the data [244]. For specific features, such as roads, hand-crafted feature extractors still produce promising results. However, as hand-crafted features rely on fixed operators, they are sensitive to variation and noise [245]. Where samples exhibit similar spectral values, techniques like K-means clustering, SVM, LDA or GMMs are utilised to find classification boundaries [246, 247]. Despite their popularity, these pixel-level classifiers struggle to model spatial context, leading to salt-and-pepper noise within their clusters or classifications [248].

Convolutional neural networks (CNNs) have become the dominant approach for extracting features from remotely sensed images. As reviewed in chapter 3.2, early studies commonly leveraged intermediate activations from pre-trained classification networks such as VGG or ResNet to represent image content. Although these "off-the-shelf" features enabled scene classification and retrieval, they also introduced domain-shift issues and often required post-processing through aggregation schemes such as SPoC, BoW, or VLAD. The use of visual words in remote sensing, however, collapses spatial information that is often crucial for discriminating classes. For example, classes such as industrial or urban residential would be difficult to discriminate with a visual words approach [249]. The codebook built would usually require retraining for each sensor, as visual words are dataset specific, reducing generalisation [250]. Most, such as Tong et al. [251], produce feature sets using common image retrieval algorithms and fine-tuning with smaller patch sizes, but based on high-resolution labelled datasets.

Unsupervised and self-supervised feature learning methods have seen popularity as alternatives to supervised CNN feature extraction. Autoencoder(AE) architectures learn latent representations directly from unlabelled inputs, see section 2.9. AEs and their variants such as convolutional AEs, stacked AEs or spectral-spatial AEs have been applied successfully across multi- and hyper-spectral data [252–256]. However, AEs prioritise reconstruction loss therefore the latent space produced may embed statistical regularities rather than semantic meaning [257]. Extensions such as variational AEs have also seen success in prediction tasks, for

example, image captioning [258], desertification [259] or biomass prediction [260]. Variational latent spaces are computationally more demanding and often require careful engineering for large remote sensing datasets [261].

More recently, masked image modelling and transformer-based architectures have been introduced as alternatives to AEs. Masked AEs adapted for remote sensing, such as SatMAE [262], reconstruct missing patches of an image and show strong performance in unsupervised spectral-spatial learning [263–265]. Masked autoencoders belong to the broader family of self-supervised methods, which learn representations from unlabelled data without relying on labelled supervision. These transformer-based methods, however, require a large computational overhead [26, 266, 267].

5.1.2 Manifold Learning

Previously explored in section 2.13 was SNE,t-SNE and UMAP, with some discussion about use cases in section 3.5.2. This section will focus on alternative manifold learning or dimensionality reduction techniques and comparisons between.

Manifold learning and dimensionality reduction share similarities, the processing of a high-dimensional input into a low-dimensional representation. Although both terms are interchangeable, reduction is the compression of data in order to keep the most valuable information. Many examples can be seen when pre-processing hyperspectral images [268]. Because hyperspectral images contain many more bands, reducing the input data size enables faster computation in subsequent algorithms. Techniques such as PCA or even autoencoders are utilised for reduction [269, 270]. On the contrary, manifold learning is the finding of intrinsic data structures and projecting into lower dimensions, therefore more suited to the envision pipeline in this chapter. Most examples of manifold learning are post-processing or towards the final stages of a pipeline.

Classical DR techniques, such as (MDS), aimed to preserve global structure, however, assume that each dimension has the same contribution. Variants include classical, and non-metric MDS, differing by whether they preserve actual distance or only their rank order [271]. Whilst MDS can reveal global structure, it is computationally expensive and is inefficient for non-linear manifolds. There have been attempts to reduce the complexity by only considering the closest neighbours such as landmark MDS [272]. Isometric mapping(Isomap) is an extension of MDS, introducing geodesic distances, making it more effective for non-linear distances [273]. However, Isomap is sensitive to noise and outliers whilst also requiring expensive

computations, all pairwise shortest paths and eigendecomposition operations. Most methods based on eigen-decomposition suffer from a problem called the repeated eigendirections problem(REP) [274]. This problem causes algorithms to fail when the manifold is, for example, a long, thin strip [154]. For many of these algorithms that were derived as closed-form eigenvalue problems, there exist implementations that utilise iterative or approximate solvers.

Local methods such as Locally Linear Embedding (LLE) and Laplacian Eigenmaps address some of these issues by preserving local neighbourhood relationships instead of global distances. LLE reconstructs each sample from its neighbours, while Laplacian Eigenmaps use a graph Laplacian to embed data based on local similarities. These approaches can capture non-linear structure more effectively but remain computationally demanding and sensitive to parameter choice. Like Isomap, they can rely on eigen-decomposition and are thus are also prone to REP.

Algorithms in this area can broadly be categorised into two types based on the type of structure they preserve. Global methods such as PCA or multi-dimensional scaling(MDS) seek to maintain large-scale relationships, whereas local methods such as isometric mapping(Isomap), local linear embedding(LLE) and UMAP focus on preserving neighbourhood relationships. This distinction is particularly relevant in remote sensing applications where both global continuity (e.g. land cover gradients) and local feature (e.g. crop types) are important.

Whilst t-SNE and Umap both surpass many of these algorithms when considering time, memory and preservation of the manifold, there still exist some pitfalls. Both algorithms still heavily rely on hyperparameter selection, such as the neighbours considered. t-SNE has a tuneable parameter "perplexity" which loosely balances global and local preservation. The algorithm, also due to the initial graph structure, can produce clusters positioning that are arbitrarily far apart. The distances between clusters cannot be evaluated as strict conclusions. Umap is also sensitive to hyperparameters; however, both of these algorithms scale to large datasets. It is for both improved manifold projection and scalability that we utilised them within this thesis.

5.2 Problem Formulation

In this chapter, we address the problem of efficient scene-level labelling of remote sensing tiles by constructing an interactive 2D plot on which a user can visually explore and assign semantic labels.

Let

$$X = \{x_i \in \mathbb{R}^{H \times W \times B} | i = 1, ..., N\}$$

denote a dataset of N RS chips, each with spatial dimensions $H \times W$ and B spectral bands. The goal is to derive a low-dimensional representation

$$Y = y_i \in \mathbb{R}^2$$

such that chips with similar scene characteristics are mapped to nearby points.

We consider a two-stage embedding pipeline. Firstly, a feature extractor to extract the latent representation in the high-dimensional space.

$$f_\theta : \mathbb{R}^{H \times W \times B} \to \mathbb{R}^d$$

Which maps each tile to a latent feature vector

$$z_i = f_\theta(x_i)$$

where $d \ll HWB$

Secondly, a manifold learning algorithm

$$g : \mathbb{R}^d \to \mathbb{R}^2$$

produces a visual embedding:

$$y_i = g(z_i)$$

Given the two-dimensional embedding Y, a user interactively selects regions of the manifold and assigns labels

$$\ell : Y \to C$$

Where C is the set of categories defined by the user (e.g. forest, coastline).

The entire problem set is to construct f_θ and g such that the two-dimensional embedding Y preserves scene-level similarity and is suitable for user labelling.

Key Assumptions:

- local neighbourhoods in latent space f_θ are semantically similar

- The underlying manifold is of low intrinsic dimensionality, enabling meaningful two-dimensional visualisation
- The learnt latent space and manifold embeddings convey evolving features

5.3 Motivation and Hypothesis

Labelling RS images is an arduous process costing time and expertise. Many methods currently rely on labelled data in order to classify unseen images, with most models unable to adapt or generalise to different geographical locations due to the variation of the Earth's surface. New datasets require a balance between inter- and intra-class variance to be able to suitably train new models for classification. Variance is hard to model with current systems, such as image retrieval or recommender systems. In addition, refinement via iterative feedback loops is hindered by the uncertainty of labelled data, even by expert users. Therefore, we have the following motivations:

- Fully unsupervised pipelines removing the need for expert analysis and initial iterative refinement
- Generalisable pipeline that can be utilised for multiple features
- Summarise datasets into an intuitive two-dimensional space for quick and effective labelling that models inter- and intra class variation
- Improve the speed at which labels can be assigned

A CNN autoencoder was chosen for this initial stage of research and the first pipeline. The model choice was predominantly made to allow for faster training with less computational overhead. Based on previous literature, UMAP and t-SNE are the most scalable and robust manifold learning algorithms to implement, especially considering the large datasets of remote sensing. Based on these aims, we have the following hypotheses:

- H1: The latent space Z learned by the AE has coherent semantic structure in which visually similar remote-sensing chips occupy neighbouring regions.
- H2: Non-linear dimensionality reduction methods produce two-dimensional embeddings that preserve neighbourhood relations in latent space Z, enabling meaningful visual exploration.

- H3: User interaction with embedding space Y is comparably faster to label images than common retrieval methods.

5.4 Materials and Methods

5.4.1 Materials

We use the public Sentinel 2 water edges dataset (SWED) [14], which contains multiple coastal geographic locations, comprised of 16 labelled $10,980 \times 10,980$ tiles (the large Sentinel 2 images are known as tiles). The images were selected to have minimal coverage of clouds with no preprocessing apart from bottom-of-atmosphere correction [165]. This dataset is rich in contrast and variety of physical coastal features in each geographical location with the addition of multiple fine-grained physical features such as jetties and bridges. The hypothesis is that we can see the transition of features and the evolution of the manifold with respect to both small features and larger geographical changes. Including water and land also introduces the complexity of the feature transition between both. Most papers within the domain of satellite imagery utilise datasets that are geographical location- or feature-specific. For example, some datasets utilise a singular tile, restricting their study to one geographical location rather than any established dataset. Most papers additionally only consider one feature from their geographically selected tile, such as lakes or forests, which will only have limited variations present in that geographical location. This geographical and feature-specific selection is also expressed by the authors of the SWED dataset [14], who also find that water and coastline mapping is restricted to singular regions in the literature as one of their motivations to publish a dataset rich in variety. As the dataset contains a wide range of land and sea features from differing geographical locations, the chosen data set is suitable for testing our pipeline.

5.4.2 Methods

We use the pipeline as depicted in Figure 5.2 where after pre-processing, we train an autoencoder to provide the latent features on which we apply dimension reduction to produce an embedding space that supports an interactive user interface.

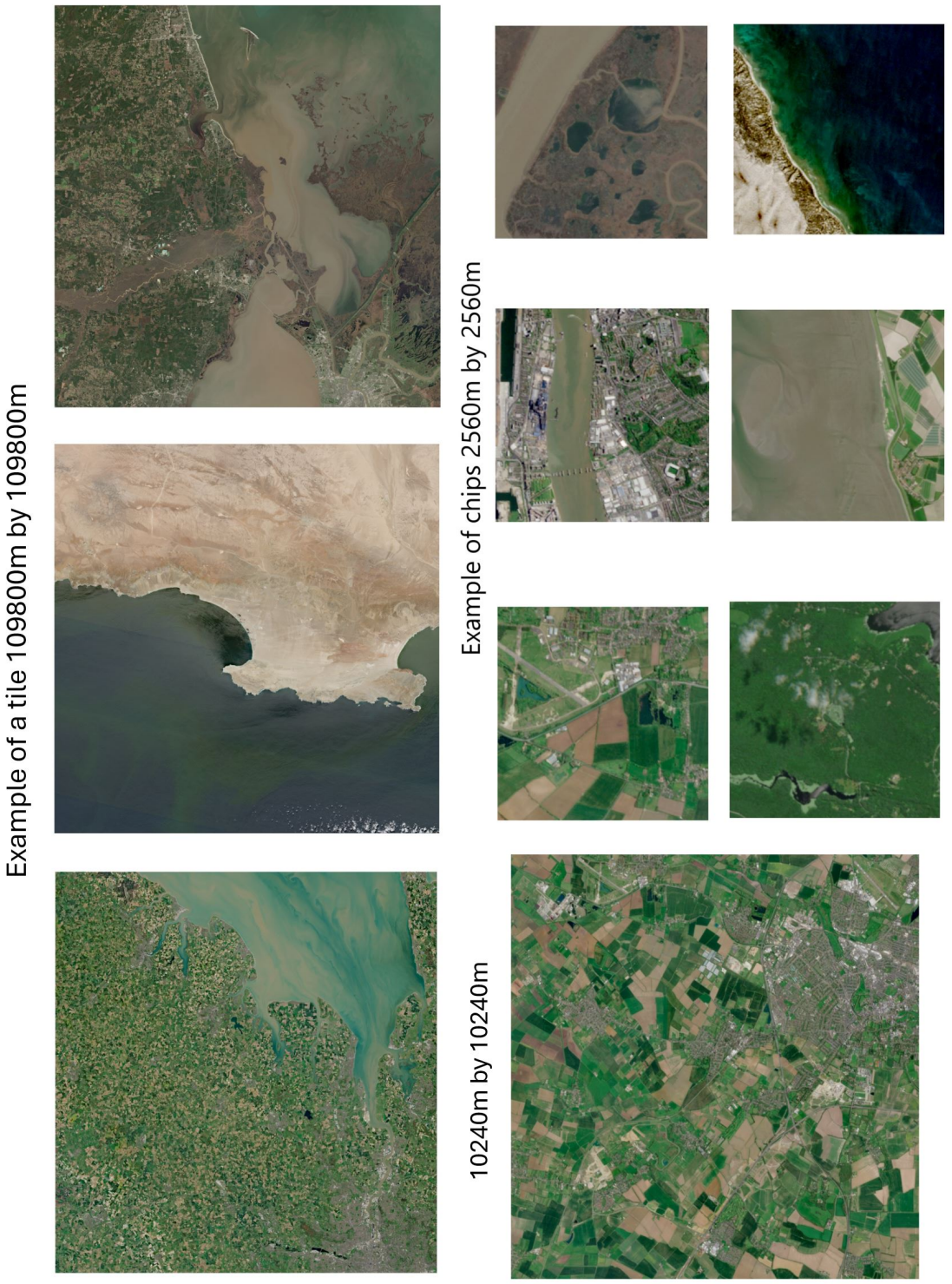


Figure 5.1: Example of Tiles spanning $109800m^2$ and an image spanning $10240m^2$. Finally, 6 example images show what a chip, $2560m^2$, contains. Chips are commonly utilised throughout this thesis.

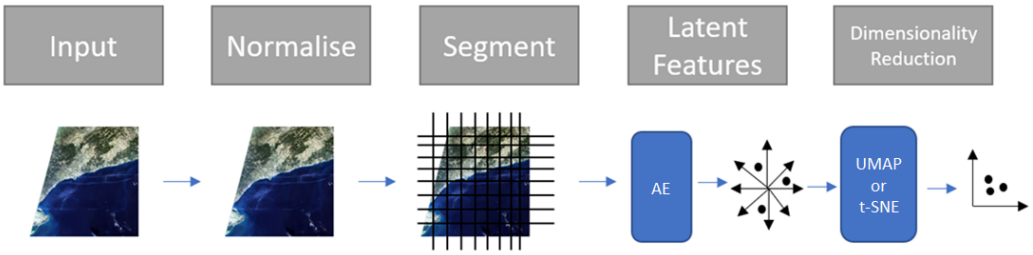


Figure 5.2: Proposed pipeline.

5.4.2.1 Preprocessing

Each tile extracted from a Sentinel 2 product is $10,980 \times 10,980$ at 10 m resolution, meaning each pixel represents 10 m^2 of the Earth's surface area. We divide each image into 256 by 256 chips to balance computational efficiency and local feature representation. For an example of chips and different scale images, see 5.1. As we do not utilise a sliding window approach, there is a remainder of 228 discarded pixels along the right and bottom border. Whilst a sliding window approach would remove edge case feature exclusion, the training time and user interactivity would be significantly more complex. Opting for smaller patch sizes could increase the performance of the AE as fewer geographic features are present. While relatively large, the selection of patch size encapsulates enough local context to be discernible without the need to reference neighbouring patches and achieves a good visual representation within the final visualisation tool. The final dataset comprises 1764 images for each tile and a total of 28,224 images across all 16 locations. Sentinel 2 products are recorded in 13 different wavelengths all at varying spatial resolution. In our analysis, we focused exclusively on the red, green, blue, and near-infrared bands as they all record in the highest spatial resolution of 10m.

5.4.2.2 Autoencoder Architecture

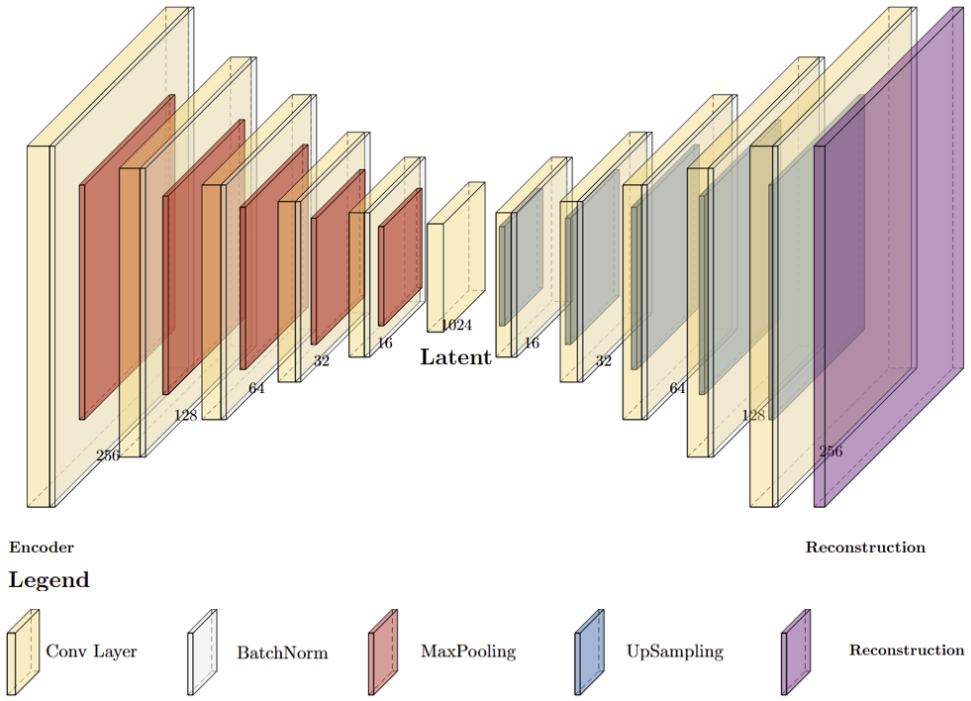


Figure 5.3: Architecture for the autoencoder. Yellow rectangles represent convolutional layers. Grey rectangles represent batch normalisation layers. Red represents max pooling layers and grey represents up-pooling layers. Purple is the reconstruction.

Each encoder within this architecture consists of a convolutional layer followed by batch normalisation and max-pooling. Stacking these encoder blocks together forms a Stacked Autoencoder (SAE). The convolutional layers use a 3×3 kernel and ReLU for activation. At the end of the encoder is a dense layer for the conversion of feature maps to vectors, enabling the extraction of the latent space variables. The decoder combines up-sampling within the convolutional layer. The decoder is mirrored in the number of, N, units stacked to create the full decoder, allowing the network to generate high-resolution feature maps, see Figure 5.3. The encoder and decoder are independent models and therefore share no information apart from the loss incurred during reconstruction whilst training.

The encoder consists of 5 blocks. The initial layer starts with 256 filters, halving for each subsequent layer. Each max-pooling layer scales the image by a factor of 2. The latent space, represented by the dense layer, has 1024 units. Training data are categorized into 10 groups based on the water content in each image, where to handle the significant difference in re-

flectance between water and land features, we weight the training batches based on the water content. Water content poses the least variation in the dataset, as it is fairly uniform when considering the texture present. Limiting the number of samples used for training removes the bias towards chips that are predominantly water, presenting no additional training context for the model. We calculate the water content using the McFeeters Normalized Difference Water Index (NDWI) [275], which measures the difference between the green and near-infrared bands. As we aim to introduce an unsupervised pipeline, we chose not to utilise the labels for water in the SWED dataset. The difference between ground truth and NDWI labels when binned was similar, apart from the noise introduced by clouds. The model is trained on mostly land content and validated on a set equally balanced between water and land. Data augmentation adds further variation using rotated images and adding salt and pepper noise, adjusting random pixels either to 0 or 1, to improve model generalisation. The optimal point in training is reached when the test loss matches the validation loss for 10 epochs, with the validation loss typically being lower due to more uniform textural information being present in water. The latent space vector provided by the autoencoder of size 1024 is passed on to subsequent stages of our pipeline.

5.4.2.3 Dimensionality Reduction

Further dimensionality reduction to two-dimensional embedding space is conducted with respect to the latent space produced by the autoencoder using t-SNE and UMAP. We chose two dimensions for simplicity and ease of downstream visualisation and interaction [276]. We use the Barnes–Hut approximation of t-SNE for its much faster computational advantages [161]. In addition, we implemented the algorithm with variable tail distribution in low dimensions to both overcome the overcrowding problem and the ability to make it heavy-tailed for tighter clusters. Our implementation enables user interaction with each t-SNE iteration step by recording each iteration's embeddings. The user can navigate between optimisation steps for any patterns that are presented earlier [240]. Whilst embeddings are not fully optimised in early iteration steps they may present dense coherent cluster grouping before later iterations optimise finer similarities between points. Data can be labelled at any optimisation step, with those labels then available to previous and future iterations. The first process of t-SNE is to optimise the embeddings with a lower learning rate and once it achieves an embedding that does not change for a few iterations, variable on data content and size, the learning rate is increased to expand and optimise the embedding. Visualising earlier iterations of t-SNE would allow a user to explore any emerging clusters without needing to wait for the finished process. As time

complexity increases with dataset size earlier iterations may provide expedient clusters to the user. Finally, we added the ability to change the perplexity parameter of t-SNE for the user to assign, depending on whether they require more global or local attention. We defined a custom version of t-SNE based on existing libraries for implementation. The resulting Python scripts are embedded into our application.

We do not make any changes to UMAP and therefore refer to the original paper for the implementation [277]. We do however allow the user to input parameters such as minimum distance and the N nearest neighbours that UMAP should compute. Both parameters govern how the initial fuzzy graph is computed by UMAP and therefore can change the visualisation and its ability to show coherent clusters or manifolds.

5.4.2.4 Visualisation

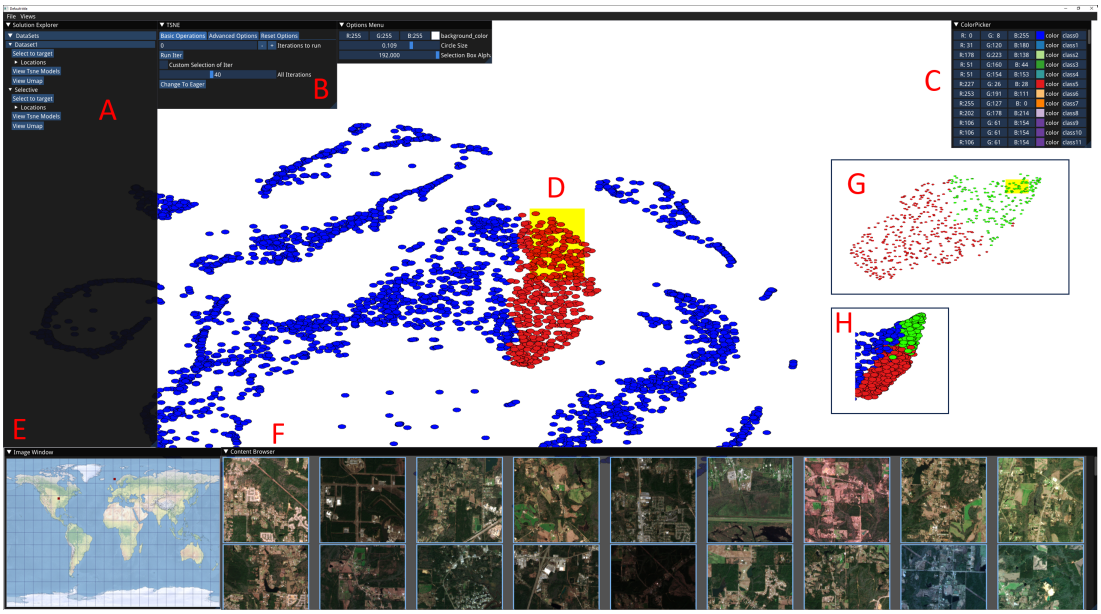


Figure 5.4: Our proposed visualisation tool for navigating satellite imagery datasets. Left of the view is the content explorer (**A**) with the ability to load multiple datasets and navigate multiple views. The controls to view the t-SNE parameters and iterations are labelled (**B**). Class colour and label control, (**C**) is housed in its own contained GUI. The main portion of the screen, (**D**) is dedicated to exploring the scatter plot. Map, (**E**) in the bottom left, views the geographical location of selected samples. The content explorer, (**F**) shows the original images for each selected point in the view (**E**). (**G**,**H**) Demonstrate class labels (colours) applied to data points.

In order to design a functional and practical visualisation application, we followed Shneidermans' well-established hierarchical set of design principles: overview, zoom, filter, details-

on-demand, relate, history and extraction [278]. These design principles were chosen because they align directly with our motivations, enabling fast and intuitive labelling and summarising datasets into intuitive manifolds. Each principle translates into a specific requirement for the tool; providing global structure (overview), enabling focused exploration (zoom/filter), and supporting annotation (details-on-demand). We also adhered to the larger ideologies present in the visualisation community such as multiple coordinated views.

Figure 5.4 presents the graphical UI for exploring the data and applying class labels to the satellite tiles. The main feature (D) is the two-dimensional embedding of the data in the form of a scatter plot that directly satisfies our goal of summarising a dataset into a two-dimensional space by forming an overview of the dataset. A content browser (F) displays the 256 by 256 image patch associated with each point. This browser is coordinated with the selection tool, a box highlight interacting with the scatter plot, for details on demand. This allows for fast interpretation of the samples without cluttering the two dimensional plot. A map view (E) plots the location of each tile supporting the relate principle by connecting embedding space similarity to real-world structure, a requirement from the domains location dependence.

Other windows (A-C) display, respectively, the data set explorer, t-SNE parameter interaction (in this case) and the class labels. Structuring these controls into coordinated windows was a deliberate choice to reduce cognitive load and maintain interpretability. Once the user is satisfied with the projection, they can minimise or close the parameter selection windows, focusing user attention on labelling. The t-SNE panel also utilises the history principle, as users can explore intermediary iterations to understand how clusters evolve, which reveals the stability of neighbourhood relationships.

After a data set has been chosen, the initial view is displayed. For each dataset embedding, there are two forms of visualisations presented to the user, UMAP or t-SNE, each can be navigated from Manifold Explorer. Each embedding in the two-dimensional scatter plot shows the overview of the manifolds and patterns within the high-dimensional space for the dataset provided. Points between clusters or within a line show the transition and relation between similar images as the features evolve for the particular manifold. For t-SNE embeddings, we included the ability for the user to customise the parameters and iterations displayed within its own contained user interface. Each embedding is accessed by interacting with a slider to swap between iterations for any extra details.

The ability to zoom and filter regions of interest is implemented with a box selection method. The box can be drawn to any size and translated on the x, and y-axis by dragging. As

points are selected, the content browser and map are updated. As all images are georeferenced, the map will have a small red marker in the geographical location of all selected samples for added context to labelling decisions. The content browser also updates to show an RGB composite of the selected images, which is how the user understands the contents of the manifold. Given the small pixel resolution of satellite images, we also included two sliders to change the image size and padding between samples.

Labelling is also conducted through the selection box. Selecting scatter points and pressing a number on the keyboard will assign all points to the respective label. Keyboard interaction is motivated by the need for fast and quick labelling, mouse selection would not utilise both hands for maximum efficiency. Selecting an image in the content browser will display a submenu where the user can select the label for the specific image, enabling granular corrections whilst still adhering to the details-on-demand principle.

We also implement an analogy of version control where a dataset can be branched into multiple subsets, each in a sand-boxed environment consisting of the same interface. The user can explore and label the subset and then merge the branch back into the original dataset with an interface to choose the labels to merge. The ability to merge branches back into the main dataset embodies Schneidermans history and extraction principles by preserving the provenance of the labelling decisions and enabling selective labelling.

5.5 Results

Accurate classification of remote sensing satellite images plays a crucial role in various applications, such as mapping, urban planning, and environmental monitoring. Key to the creation of robust and accurate image classification algorithms is the creation of large, geographically distributed labelled datasets that represent the complexity and diversity of the Earth's surface. The creation of such labelled datasets is time-consuming, resource-intensive and uncertain, as it is difficult for remote sensing analysts to easily quantify and understand the complexity and diversity present within a geographically distributed unlabelled dataset.

Labelling satellite images is necessary for both producing up-to-date maps and creating new labelled datasets for training new models. Finding selective images that contain desired features for a new training dataset can be a daunting task considering the volume of tiles over time and geographical location. Training models on satellite images incurs a high cost in both training and finding suitable data to train on.

The inputs to our labelling application are the satellite images that need to be given class labels, a trained model (e.g., the autoencoder we use for testing), and any prior labels (e.g., saved from a previous labelling exercise). The output will be class labels for the images.

A key point is that the pre-trained model may provide good cluster coherency, but often there will be out-of-class samples that will negatively impact the labelling experience and hence the time to undertake labelling. Our interface helps this by allowing the user to explore the manifold by creating user-directed two-dimensional embeddings from the high-dimensional embedding. The user can control the development of the embeddings, use the branch and merge feature (discussed later), switch between embedding types (UMAP or t-SNE) and extrapolate class labels forwards and backwards through embedding iterations.

The case study is provided in cooperation with the UK Hydrographic Office. Our approach can be used with various applications in mind. For example, to quickly label similar samples we aim to bring large numbers of tiles with similar features together in the interface to apply a single class label in bulk. Another example would be to build a data set with a good distribution of rich and diverse features, where in that case we may focus on cluster boundaries to provide images with combinations of different features and differences from the cluster.

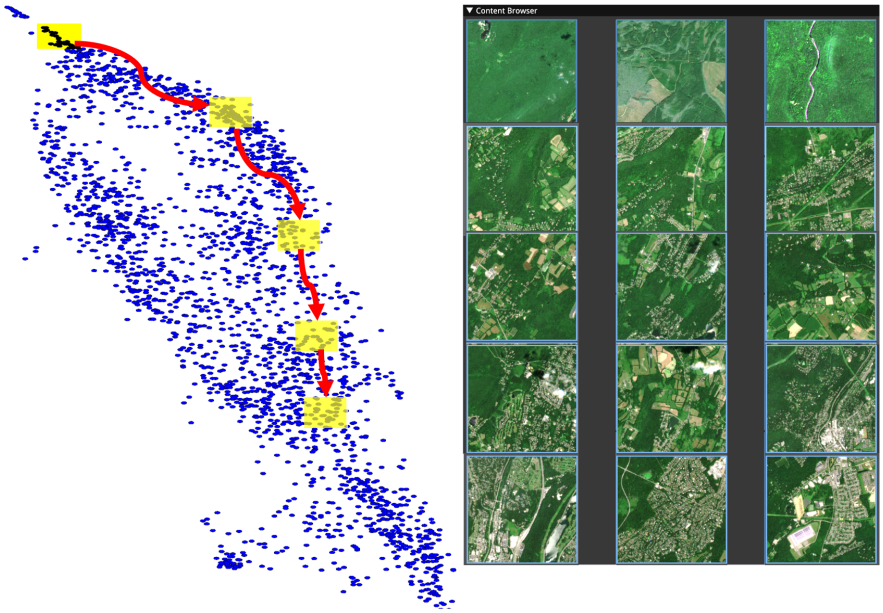


Figure 5.5: Example of images evolving from forest to dense urban whilst traversing through a manifold. Starting from the top of the two-dimensional embedding, subsequent samples were taken while moving the selection down. The respective contents of the samples are shown on the right. The accompanying video provides a better understanding of manifold evolution.

Manifold exploration: We explain how the user interacts with a two-dimensional embedding to gain an understanding of the higher-dimensional manifold and to label multiple images simultaneously with a class label. The user draws a selection box of a desired size over the plot (yellow box in Figure 5.5). Each point within the selection corresponds to one image, which is displayed in the image browser and located on the map. All images within a selection can be given the same class label. If any images are not part of the class, they can be individually reset or set to a different class using the image browser. The user can navigate the scatter plot by scaling and translating.

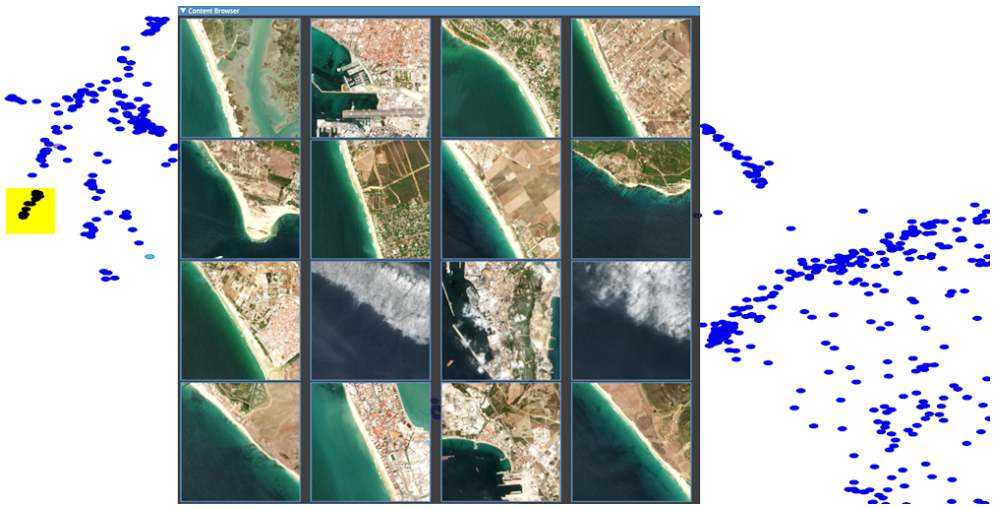


Figure 5.6: A cluster of points is seen to be mostly coastline, but the model has placed two cloud images, which have a similar reflectance and texture, in the same cluster.

The user drags the selection box so that points exit and neighbouring points enter the box. This simultaneously updates the image browser and map view, thus allowing the user to explore and become familiar with the structure of the manifold. Fine movement can help produce views where all images are consistent with one label, allowing a single label to be rapidly given to all points in the selection. In Figure 5.5, the user follows a path as indicated by the red arrows and yellow boxes, and sample images along the manifold from within the yellow boxes are presented in rows respective to the sampling region. These demonstrate how this model and two-dimensional embedding have placed images from the manifold. The block of images on the right shows three representative images from each of the indicated selection areas demonstrating the smooth evolution of the manifold from forest to urban land cover. Figure 5.6 shows 16 images in the browser window over the top of the scatterplot view. The images are from a small cluster on the left and show largely consistent clustering, but also demonstrate

how the limits of a pre-trained model can affect labelling software. In this case, the reflectance and texture of cloud images have not been separated from the reflectance and texture of similar coastlines. The interface allows these labels to be quickly corrected individually. Users can label the data efficiently with a numbered key input as they explore the manifold. Numbered key inputs limit the user to ten classes at a time, considering each digit from 0 to 10.

Manifolds shown in the two-dimensional plot are formed by both the AE's expressiveness on the source images and the number of samples that the DR technique is provided with. When projecting all images from the SWED dataset, there is a clear separation between land and water clusters, but images containing both features (coastline) are on the edges of those clusters or form paths between the clusters. These results partially support $H1$. Regions of the latent space show smooth semantic transitions(Figure 5.5), but globally the hypothesis does not hold, for example the cloud clustering with the coastline(Figure5.6). See the video in additional material for an example of coastline labelling.

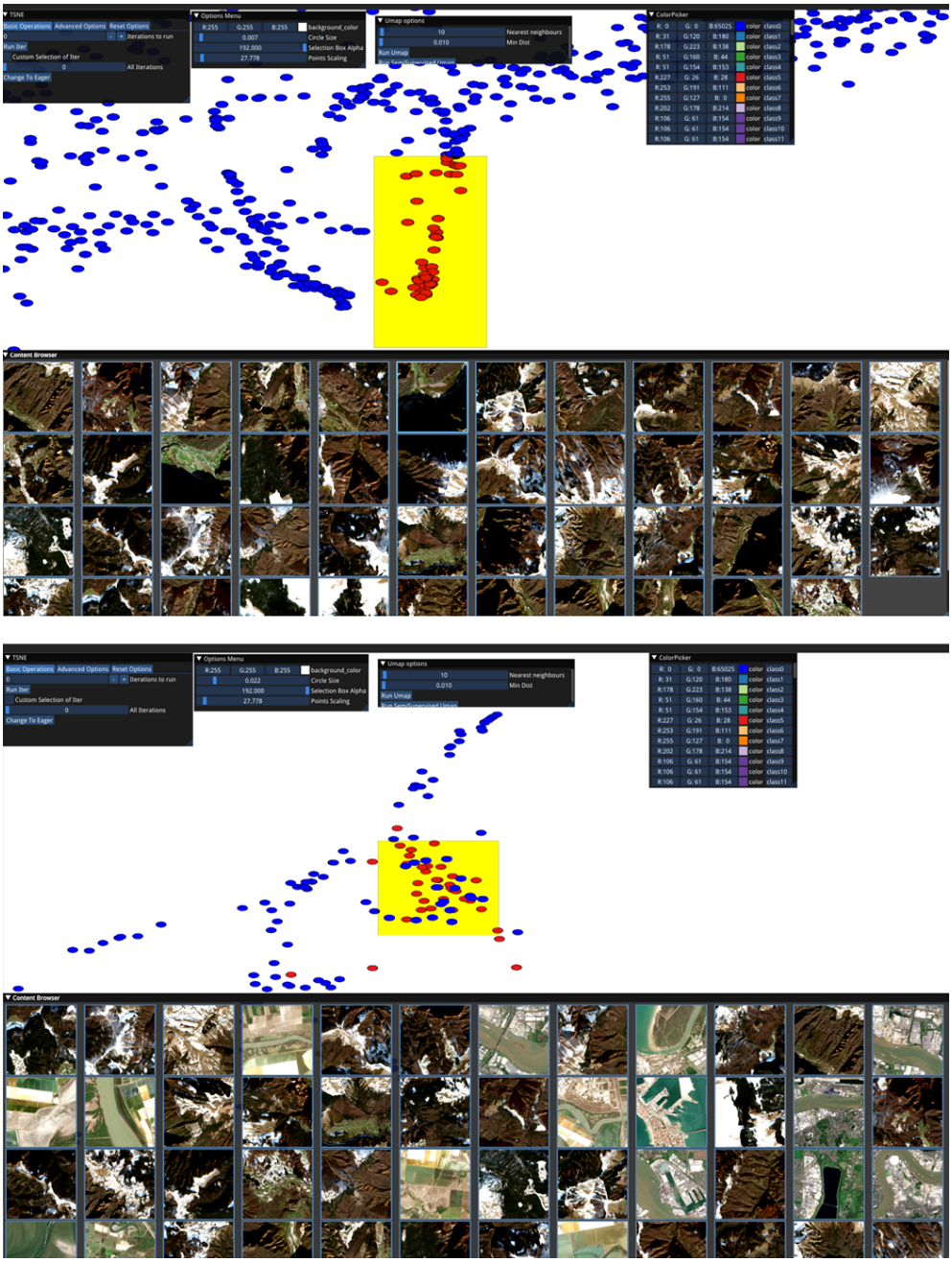


Figure 5.7: In this particular case, t-SNE (**top**) effectively clusters mountainous regions compared to UMAP (**bottom**) using the same embedding.

User-directed reprojection of the embedding: The two-dimensional scatter plot view may not be optimal due to two reasons. Firstly, the model's latent space may not be expressive enough. Secondly, which is the focus of this section, the dimension reduction tech-

niques have insufficiently exploited the latent space to successfully create the two-dimensional embedding($H2$). We work under the assumption that the AI model is fixed due to the fact that training remote sensing models requires a tremendous amount of resources, and therefore we cannot alter the latent space interactively. Rather, we look to optimise the embedding instead.

To find a better embedding we allow the user to direct re-embeddings by testing different parameters such as perplexity for t-SNE or N neighbours for UMAP. Increasing the value of both parameters allows for the DR algorithms to consider a larger neighbourhood, which provides more context to each embedded point. Conversely, if two manifolds exist in high dimensional space that overlaps frequently, a smaller neighbourhood of points would be preferable.

Therefore, it is crucial that the user can interact with parameter selection (B in Figure 5.4). However, recalculation of the embeddings on the entire dataset requires a high computational cost and exploration cost as new embeddings require validation of information patterns by the user. Constant evaluation of newly projected manifolds can be mitigated with our tool because labels can be extrapolated when switching embeddings (between UMAP and t-SNE) and when altering any of the parameters. This consistency in class labels can provide the user with context, e.g., how previously neighbouring points may now cluster or spread across the updated embeddings. In Figure 5.7, t-SNE (top) has clustered mountainous images (with glacier), allowing the user to apply a label with a single key press. Below, Figure 5.7, UMAP (bottom) has not differentiated such images from other terrain. By labelling using t-SNE, and then changing embeddings, we can see how the labelled points redistribute and mix within other samples. It is not generally the case that t-SNE performs better, rather being able to switch embeddings or re-embed using new parameters allows the user to find appropriate clusters. Whilst the two-dimensional embeddings preserve several neighbourhood relationships from Z, supporting $H2$, it is dependent on the manifold learning technique, which can introduce unwanted mapping, as shown in Figure 5.7.

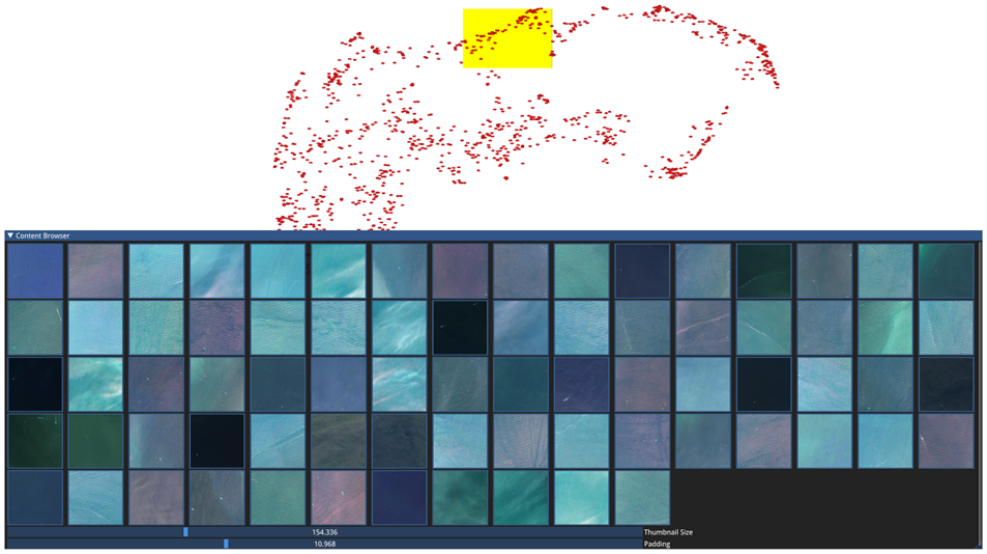


Figure 5.8: Left points in the left of Figure 5.6 are branched out into their own embeddings.

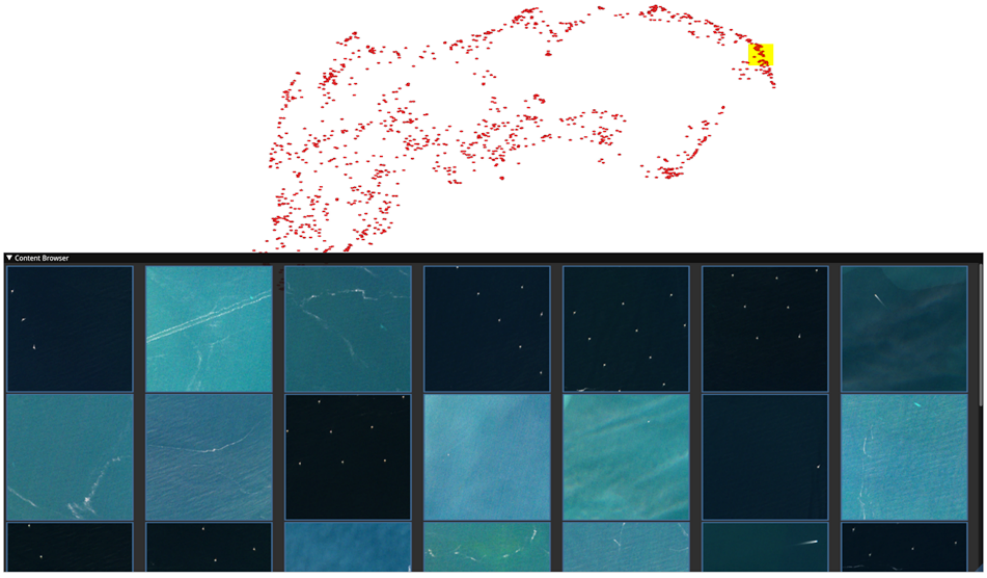


Figure 5.9: To the right of the plot from Figure 5.8 there are some images of wind farms.

Branch and merge: As an alternative to global embeddings, we introduced the ability to branch the dataset. Branching focuses the dimensionality reduction on the currently selected data. The two-dimensional embedding is optimised with respect to those samples as only those samples are considered. The computational cost is less due to the reduced number of samples when searching for optimal embedding parameters. This functionality aims to overcome the

limitations of manifold learning techniques, whilst also trying to support hypothesis $H3$, faster labelling. If the current hyperparameter selection is suboptimal or the choice of manifold learning, the user can simply reproject affected points rather than the entire dataset. The data for a particular branch is determined by the user indicating which class label or classes are to be used for that branch. All images labelled as that class form the branch. The user may select data directly on the two-dimensional plot using the selection box, apply a class label, and send that set of points to a branch. To self-contain branches, all embeddings utilise the UMAP or t-SNE algorithms for optimisation in their own sand-boxed environment without affecting any other dataset or branch.

Working with this reduced set of samples has two benefits. First, embeddings are faster to compute when varying parameters during exploration. Secondly, as the samples will have been determined by the user to have similarities, re-embedding them will effectively amplify the remaining feature differences yielding new clusters based on the alternate features. The new embedded clusters may separate better than the original given the same pre-trained model output.

Figure 5.8 shows a branch of data from Figure 5.6, which largely represents images with only sea and varying coverage of coastline. In the global embedding, they clump together in a smaller cluster in which it would be difficult to make selections that separate these classes for labelling. Branching out these points allows a re-embedding of the points, resulting in the embedding seen in Figure 5.8. In this case, the points scattered along the top of the new plot are predominantly the images that contain just the sea, allowing the user to make large selections of tens to hundreds of images and directly apply one class label.

Moving along the top of the plot and to the right (Figure 5.9) shows some wind farm feature where we can see a regular grid of white turbines against the dark blue sea suggesting the user would be able to quickly label sea infrastructure as a separate class if desired.

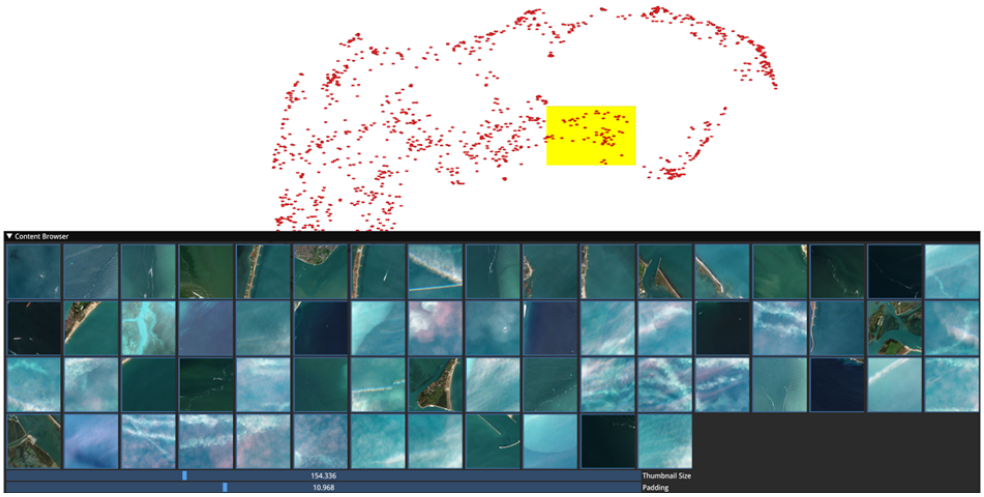


Figure 5.10: A cluster of high cloud,sea and coastline.

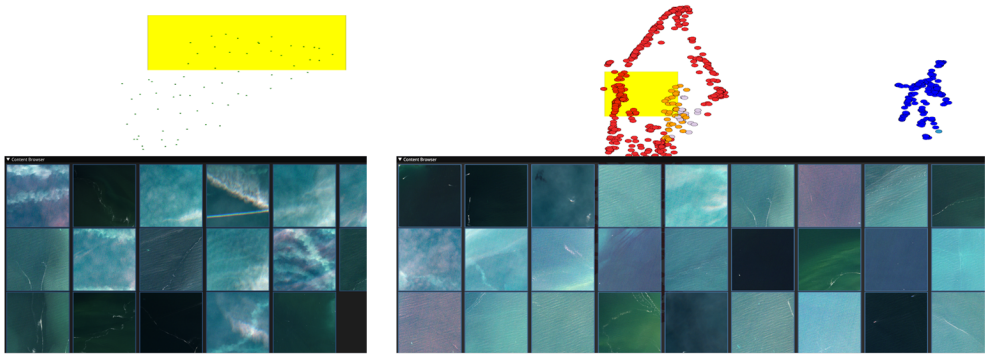


Figure 5.11: Left, a branched embedding allows the user to quickly label sea images. Right, the labels (of both coastline and sea) are merged back into the original embedding. The newly labelled images from the branch are orange and grey within the original embedding. Guided by the labels, the user can make a large selection of sea images which can be labelled with one click.

Further around the plot (Figure 5.10) there is a mix of coast, high cloud and sea that has not been effectively separated and therefore would be time-consuming to label. This can also be branched out into its own where the sea is more effectively separated in this new embedding (Figure 5.11 (left)). Labels are then merged back into the main embedding (Figure 5.11 (right)) showing the separation (or lack of) between these classes. This allows fast labelling of the data, which also allows the user to visualise the class boundary, allowing an assessment of model performance by suggesting where the model may be failing to provide good class separation.

We also find that branching is effective in another way. If a user wants to label patches with multiple features, for example, coastal areas with farmland present, they can branch and embed

all coastal patches together with a selection of patches that contain only farmland. The embedding will create unique clusters for each feature; however, samples that have both features will lie on the edge of the cluster closest to the cluster that contains the other feature. In this case, samples with both coastline and farmland will be on the edge of the coastline cluster closest to the cluster containing purely farmland. In this way, branching can act as a way to query image features in clusters by leveraging the model's general ability to distinguish between them.

5.5.1 User Study

In this section, we introduce a second application (Figure 5.12) to act as a comparison in a user study. We compare the efficiency of finding coastal patches utilising an image retrieval application as a benchmark, as our pipeline can be interpreted as a visual extension of image retrieval systems. The benchmark application produces K similar images based on a query image. The K nearest neighbours are determined by distance in the autoencoder latent space. The target query image is shown with the most similar K images. The user can label all images returned by the query or individually label each sample.

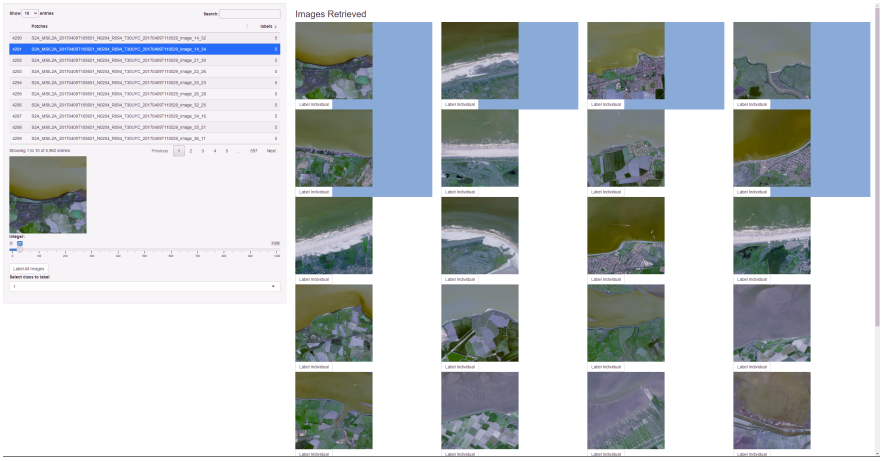


Figure 5.12: Benchmark imageretrieval application. The selection of query image and K images to retrieve is located on the left. The main panel displays the resulting images returned from the query.

To access the capability of each application we conducted a small user study with five participants. The user study consisted of each applicant labelling patches with any coastline utilising both applications. We presented the participants with forty initial "seed" images, as otherwise, the benchmark application required sorting through a list of 5963 patches to find an initial coastal patch to begin labelling from. The same "seed" patches are pre-labelled within

our application as one class.

The task was to label 160 patches with coastline using each application. We allowed participants to label more patches than required accounting for any scenarios where labels were assigned with the intention to refine after. The method to end the study was based on a small GUI element that let the researcher know that the labelling requirement had been met, allowing them to make an informed decision on when to let the participant stop labelling. We switched the order in which both applications were presented to each user in order to keep any familiarity gained with RS images consistent. As both t-SNE and Umap are stochastic algorithms we retained the initial randomisation seed between participants. In addition, we removed multi-threading and approximations utilised by the dimensionality reduction algorithms in order to keep consistency between participants. Times were recorded for each patch labelled. In order to produce the results, any patches labelled multiple times were removed with only the final label assignment considered.

The results of the user study are shown in Figure 5.13. On average, the time taken to label using the benchmark application was 269 s compared to our application's 71 s, which is a factor of more than 3 times faster. Users utilising the benchmark application often found that only a small number of patches could be retrieved without the appearance of out-of-class images, which reduced the ability to label all samples quickly and efficiently, resulting in single-sample labelling. Label-times decreased for all participants, supporting $H3$. Variation between users suggests a learning-curve dependency rather than uniform performance gains. In contrast, in our approach, the ability to optimise the box size and explore its surroundings produced similar images within the embedding, allowing the user to discern a suitable selection, which retained minimal out-of-sample patches and allowed the user to refine the selection with more confidence. The fastest user strategy recorded utilising our application searched the space before selecting clusters with more coherent samples and only a few outliers. In contrast, the other common labelling strategy used was panning a small selection area with continuous movement and refinement. The difference can be seen in Figure 5.13. The sharp increase in labels corresponds to labels applied to a larger selection area at once in contrast to the smoother gradient when labelling with a small selection strategy, respectively. It is also noticeable that users of the benchmark application experienced fatigue as they approached the target and had difficulties finding more coastline to label, whereas users of our application did not experience that problem. This is visible in the shallower gradients in the data for the benchmark application as time goes on, compared to the steeper gradient for our application.

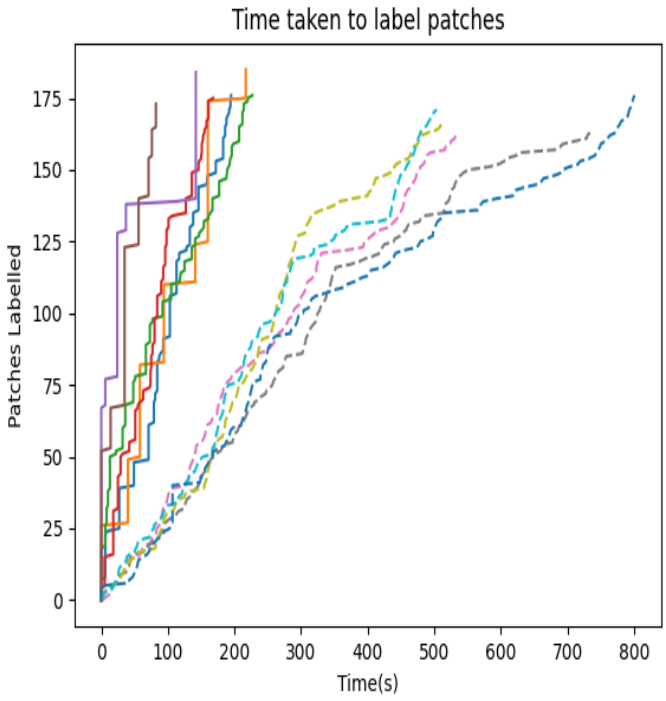


Figure 5.13: Line chart showing the difference in participants' time taken to label coastline patches. The dashed lines represent times taken in the benchmark application. The solid lines represent our application. Each colour represents a different participant.

5.6 Discussion

With respect to our first hypothesis $H1$ we found that there is a meaningful semantic relationship between chip in latent space Z. This result is despite not utilising masked autoencoder models that provided semantically improved latent spaces within the literature. This relatively lightweight implementation is suitable for scene-level exploration and labelling. There is a clear separation between land and sea features, which is to be expected given the large discrepancy between textural features of those classes. The resultant manifold projection of the entire dataset embodies this separation, with land features notably on one side and water on the other. Within land features the manifold can be seen clearly evolving from vegetational areas to more urbanised environments. The presence of man-made features is seen to be in stark juxtaposition to vegetation by the AE, the encoding of individual chips into Z is dependent on the ratio of these features. Likewise, within samples that contain a majority of water, there is some evolution between offshore windfarms and ships compared to solely sea scenes. Whilst

this evolution of features between man-made objects and natural features is present for land and sea chips, it is less pronounced for man-made objects in water, as they are usually sparse and smaller. The embedding of points between both major features, land and sea, is not only occupied with coastal, rivers or lake chips but also with those with sharp textural changes.

The large textural changes in this region are some of the outliers we have found directly opposed to our hypothesis $H1$. Outliers within the latent space Z, namely the inability of the model to distinguish between cloud and land coverage when present next to water features. This discovery is notable not as a commentary on the reconstruction loss, however, an effect of AEs inability to encode semantic meaning, in favour of statistical regularities [257]. As the evaluation of the model is with respect to the embedding space Y, there is no explicit evaluation of the feature space Z; manifold learning techniques are used to infer meaning from the feature space Z. We discuss this limitation within the following section 5.7.

Regarding the second hypothesis $H2$ we have shown that it has been partially held when considering only one dimensionality reduction technique. Notably, when utilising UMAP, the embedding of mountainous regions was shown to occlude with other water-based features. The result was not due to the latent space Z however from the embedding technique utilised, the t-SNE embedding produced optimal embeddings. In order to alleviate the problem we incorporated a branch and merging feature. Branching enabled the user to reproject a subset of samples in Z utilising different parameters or techniques. Reprojecting a small subset, opposed to the entire dataset, allowed for on-the-fly embedding calculations improving labelling speed and reducing the limitations. Branching a subset of the data, however, removes the global context of the data; no relationships can be inferred apart from those within the subset. The utility of the branching feature is primarily for addressing sub-optimal embeddings; however, it was also found to amplify feature differences. Where only a subset is utilised, when branching, the manifold learning technique can optimise toward the differences found only within those images.

Most importantly, our user study directly supports our final hypothesis $H3$. User labelling utilising our tool shows notable improvements to labelling speed in comparison to existing image retrieval-based systems. The ability to infer semantic meaning from the evolving manifolds enabled users to focus their attention directly on samples required for labelling. Whilst the latent space Z doesn't explicitly encode semantic meaning, the learnt representation can still infer meaning to a human user. The users tasked with labelling coastline found coastal samples between the land and sea feature, ascertaining an understanding of the quantity of

coastal chips and their positioning in the manifold.

5.7 Limitations

There are two sets of limitations to the current pipeline, those inherent to the design and those found in practice. Inherent limitations include the granularity of labelling or the lack thereof. As this work is aimed towards scene-level labelling and classification, smaller features are lost in the dominant background features. For samples with only two features one dominant (ocean) and one smaller (windmills) our approach still allows for efficient labelling. However, discerning and labelling multiple smaller-scale features like those found on land(roads, dirt roads, sheds, houses, motorways, etc.) cannot be as easily separated into respective categorical labels if they are all present in the same chip. From the labeller's perspective, it is much more efficient to label the background characteristics rather than more granular features, due to the use of scene-level encodings.

The second set of inherent limitations is introduced by the use of dimensionality reduction algorithms. Whilst we have shown their effectiveness at showing evolution between images, there are still major drawbacks. One such drawback as explored is the differences in methods, t-SNE or UMAP, produced either desirable or undesirable results, see Figure 5.7. Different dimensionality reduction methods produce embeddings with varying levels of cluster continuity, no single method is optimal across all datasets. In addition, there are scalability constraints when considering extremely large datasets. Embedding can be pre-calculated offline before human interaction however this doesn't account for re-projection if different hyper-parameters are wanted. Scalability and time constraints of embeddings also affect the branching methodology if and only if the user selects a large number of chips to subproject. For our results, we implemented the solution on a system running an Intel i7-11700k, which took roughly 9 minutes to project all $28,224$ chips. The current dataset, whilst large, could be considerably larger considering the needs of RS.

In practice, we found that there is some semantic leakage between scenes that share the same texture. A notable example of semantic leakage is the inclusion of scenes with cloud coverage among scenes containing coastal images; see Figure 5.6. The semantic leakage also highlighted another limitation, textural orientation was more pronounced within the embeddings. Textural changes that share an orientation are considered more similar. Similar scenes should not be placed within the manifold based on textural orientation, as that negatively influ-

ences the manifold's construction. The manifold can become fragmented or discontinuous if image orientation is more salient than image content.

Lastly, this work was evaluated with more qualitative metrics rather than quantitative. Qualitative evaluation requires human input which can be costly. Solutions should consider the use of quantitative metrics that can primitively assess potential issues such as the semantic leakage. Consider a metric that can assess sample similarity within an embedding compared to its neighbours. This potential metric could select the most informative embedding from multiple methods and hyper-parameters, this approach could also highlight potential issues before human analysis.

5.8 Future Work

Common image dataset tools in current works regard finding the top K images that correspond to a query set of images. Most, such as Tong et al. [251], produce feature sets using common image retrieval algorithms and fine-tuning with smaller patch sizes, but based on higher resolution labelled datasets. We have, in comparison, shown the feasibility of utilising manifold learning techniques to enable entire datasets and relationships between images to be displayed. This has been effective in mass labelling larger feature sets. When considering smaller local features, we utilise our branching methodology. Future work could also encode smaller patches with extra information regarding surrounding patches, increasing the focus on smaller features, but retaining a more global context. This would have to be balanced between computational efficiency for embeddings, as larger latent spaces or sample rates increase the time complexity to calculate embeddings.

In comparison, for better potential label separation, we could iteratively alter our higher-dimensional space by pushing dissimilar images from an oracle or user's perspective away and pulling similar images closer. This process in machine learning is known as triplet loss [279]. Successfully applying triplet loss improves the downstream visualisation capabilities as manifolds separate clearly. However, this process would require retraining on the new high-dimensional embedding space, and constant oracle/human input where labels can heavily influence models' class separation. With such feature-rich patches, this could cause overlap between labelling categorisation, for example, where a particular patch includes multiple regions of interest, i.e., farmland and rivers. An avenue for future work could look to implement a form of triplet loss with a multi-hierarchical labelling scheme utilising the visual branching feature we presented.

In addition, the utility of understanding the performance of a model between the RS images utilised in training and the resulting complex learnt filters by the convolutional AE is paramount in any use case. The user can gain estimations of model performance by altering the manifold learning representations and examining the visualisation provided by the projected manifold. Our methodology could be adapted to any model where a temporary layer can be trained to extract an embedding space. Limitations are the size of the representation of each image, larger high-dimensional spaces require more time to project. The resulting embeddings within the tool provide contextual information about how samples are built by the model and where problematic features may arise in images.

5.9 Summary

This chapter explored the performance of autoencoders and dimensionality algorithms to effectively embed scene-level images for user labelling. The method proved to be more than threefold faster than comparable image retrieval methods. Whilst the latent space proved to have coherent semantic structure, the ability for the algorithm to differentiate some textural features needs to be improved. Likewise, both manifold learning algorithms produced suitable two-dimensional plots for labelling; however, there is no one method that is superior. Manifolds within the embedding space can accurately convey the evolution of urban environments to more agricultural and forestry environments, with embeddings having a clear separation between land and water. An exception is large shadows present within the images, such as those cast by mountains. The large textural difference between shadows and mountainous terrain could be mistakenly embedded near coastal regions. However, utilising a different embedding algorithm created a coherent split, indicating a very close encoding but separated nonetheless. Similarly, the same problem can be seen with cloud and sea boundaries; however, this is due to the encodings, not the two-dimensional embeddings.

Chapter 6

Leveraging Convolutional and Graph Networks for an Unsupervised Remote Sensing Labelling Tool

The work presented in this chapter is under review [30] in the journal Annals of GIS with the title *Leveraging Convolutional and Graph Networks for an Unsupervised Remote Sensing Labelling Tool*. The paper has been accepted and peer-reviewed , publication is subject to edits requested by reviewers.

Chapter 5 introduced our first unsupervised pipeline for satellite image labelling, in which scene-level chips were encoded using a deep convolutional autoencoder and projected into a low-dimensional manifold. This approach demonstrated how large unlabelled datasets can be effectively navigated to identify a coherent manifold for fast labelling with minimal user effort. The application further introduced interactive branching and merging of embeddings, allowing users to refine local regions of the manifold and consolidate them back into the global view. Whilst the work showed clear advantages over conventional image retrieval tools, one clear limitation remained, the method is primarily suited for scene level labelling. Scene-level labelling restricts the functionality of our tool to a single semantic label per image.

For many remote sensing applications, coarse annotation is insufficient. Individual chips may contain mixtures of land cover types, for example, urban structures surrounded by vegetation or rivers running through agricultural fields. Many domains, such as agriculture, environmental monitoring and hydrography, require labels at finer spatial resolutions. Labels are often applied to pixels, homogenous regions or objects within the image [23]. The scene-level

embedding approach in chapter 5, while effective for broad semantic categorisation, does not explicitly model the internal structure within each chip nor the geographical relationship that influences semantic meaning. The lack of desired functionality is demonstrated by the textural orientation problem between land or clouds and sea.

In this chapter, we expand our application towards granular, segmentation-level labelling. Instead of encoding entire chips into a latent space, the chips are represented by superpixel segments. The new superpixel segmentations encodings capture both spectral characteristics and local geographical context. To achieve this, a weakly supervised pipeline is adopted, for which fuzzy C-means clustering generates soft assignments for segmenting targets. In comparison to chapter 5 where latent features were optimised based on reconstruction loss, causing embeddings to encode statistical regularities, this work aims to encode semantic meaning. Convolutional neural networks are used to extract discriminative textural features for each segment, and graph neural networks are introduced to encode adjacent relationships. The proposed pipeline ensures each segment not only reflects its own content but also the composition of the neighbouring segments. While graph-based methods and superpixel representations have been widely explored in remote sensing, as shown in the following section 6.1.2, most existing approaches focus on classification. In contrast, this chapter introduces a graph-based representation designed explicitly for unsupervised similarity assessment and segment labelling in a tool. This work addresses the reliance on ground truth data and the lack of support of interactive dataset construction.

This shift from chip to segmentation encoding addresses several limitations observed previously in chapter 5. First, segmentations allow users to assign multiple labels to a singular chip, as each chip is represented now by multiple labelled segments, increasing expressiveness. Second, the incorporation of a graph-based matching removes the outliers that arise from the dominant textural orientation problem, where chips containing clouds were mistaken for land and sea boundaries. Finally, by utilising a CNN encoding and GNN modelling, the revised pipeline produces a more structured embedding space in which users can explore and label at a finer granularity.

In this chapter, we propose an improved unsupervised feature extraction for visual labelling and exploration. **A**) We provide more coherent embedding spaces with tighter cluster formation for easier navigation utilising unsupervised clustering. **B**) With the addition of Graph neural networks and feature encoding on segmentations we show that the resulting two-dimensional embedding space is rotationally invariant. **C**) Finally, We use this methodology to create a

more advanced exploration and labelling tool.

6.1 Background

6.1.1 Superpixel Segmentation

Superpixel algorithms aim to find homogenous regions of close pixels in order to lower the amount of primitives for downstream processing, see section 2.7.3. These parcels of pixels represent natural entities, or subcomponents, within images. A homogenous set of pixels is defined by one or more metrics: colour, gradient, space or texture. Gradient change for segmenting into superpixels can be seen in algorithms such as watershed [280]. Variations of watershed can also combine multiple different metrics [281]. Colour or texture algorithms are commonly utilised for segments that contain similar elements within a single region. Conversely, gradient change is only applicable to the boundary conditions of a superpixel.

Superpixel segmentation is particularly useful in remote sensing because satellite images often contain heterogeneous mixtures of materials, and reducing pixel-level noise into coherent regions helps stabilise feature learning and improves interpretability for downstream models. In the context of this thesis, superpixels also serve as a bridge between scene-level representations, as seen in chapter 5, and the finer spatial granularity required for segment-based labelling.

Applications of superpixel segmentation can fall into the watershed family of algorithms, clustering-based, graph-based, or energy-based. As the name suggests watershed algorithms consider images as containing peaks and troughs based on the greyscale values or gradient magnitude. A sharp incline in the gradient is considered a peak. Liu et al utilised watershed algorithm with spectral clustering to segment trees within a forest from Lidar data [282]. Pure watershed methods suffer from over segmentation or seeding sensitivity. Some methods look to improve the performance via post-processing segments by merging or splitting [283]. Alternative improvements can be made by optimising markers before utilising the algorithm [284]. Commonly, the watershed algorithm is seen utilised for a singular purpose, be it forests, water features, ships, etc., due to the domain-specific nature of improvements made to the algorithm. These domain-specific optimisations highlight a broader limitation, many classical superpixel techniques require tuning or adaptation for particular material types, which motivates the exploration of learning-based feature extraction (e.g., CNNs and GNNs) to better generalise across diverse remote sensing data.

Clustering-based methodologies utilise clustering techniques such as k-means. For example, Simple linear iterative clustering is essentially a localised k-means algorithm that centres on initial markers. Making clustering techniques fast to execute and scale well to large datasets. Similar to watershed methods, they are initialised with markers. However, markers are evenly spaced and often produce compact regular shapes. Their boundary adherence can be poor: compactness bias causes leakage across irregular boundaries. Many use cases can be seen with clustering segmentation as a precursor to AI model input from vision transformers to graph networks [285–287]. Most methods rely on over-segmenting the image as these algorithms are sensitive to noise and over-reliance on spectral colour-space [288]. The reliance on over-segmentation is advantageous for this work, as it preserves fine structural details that are crucial when constructing segment-level feature vectors and graph neighbourhoods. Over-segmented superpixels provide a robust starting point for learning contextual information rather than relying solely on raw pixels.

The problem can also be modelled as a graph. Each pixel is a node and a weighted edge between neighbouring pixels, for example, the spectral similarity. Techniques in this area utilise partitions in this graph, such as normalised cuts, Felzenszwalb or entropy rate methods [289]. These methods allow for a more flexible similarity modelling. Edge weights, optionally combined with spatial context, in a graph structure can better respect non-local affinities. Additionally, some graph objectives(entropy-rate, or normalised cuts) create a good balance between compactness and boundary adherence [290]. Graph methods have two major flaws, the large memory requirements and computational expense. More recent implementations utilise cluster-based approaches to construct initial segmentations and then refine via graph methodologies [291].

Lastly, common energy-based methods optimise an energy/objective function based on homogeneity, regularity and other balancing terms. These methods iteratively find the local minima of the energy function. Common algorithms in this field are SEEDS, entropy-rate superpixels or graph-cut variants [292]. Explicitly defining the parameters within the energy function, size or compactness, allows for more fine-tuning than other methods. Unfortunately, with any local minima approach the optimal solution may not be found and it highly dependent on the optimisation of the energy function.

To summarise, segmentations provide a method to group local homogeneous pixels. Practically, clustering-based methods are the most common and effective for initial segmentation. Applying merging or refining techniques using graph-based methods after initial segmentation

improves quality and keeps the computational cost down. Refinement is normally done via computationally expensive methods, energy or graph-based approaches.

For the purposes of this chapter, superpixels form the foundational units for constructing segment-level feature vectors and spatial graphs. By operating at this intermediate level of abstraction, we move beyond the limitations identified in Chapter 5, where entire chips were represented as indivisible units and enable finer-grained labelling that preserves both local semantics and global scene context.

6.1.2 Graph neural networks and superpixel segmentation

Superpixels are a versatile representation of the original image, many of previous methods that work on images can be adapted to superpixels. Their uses can range is vast from clustering to vision transformers and graph neural networks [285–287, 293, 294]. In particular, graph neural networks are of particular interest to this thesis. Graphs have the ability to encode spectral and spatial features via feature encoding and edge weighting. We suggest that the reader refer to a survey by Zhao et al. [295]. They have summarised the utilisation of graph neural networks within hyperspectral images. The combination of superpixels and GNNs is particularly effective for remote sensing because pixel reflectance values alone are often insufficient for accurate modelling. Incorporating contextual features, the relationship between neighbouring segmentations, can significantly influence class semantics. This contextualisation was missing from the pipeline in chapter 5, motivating the pursuit of graph-structured representations.

Most applications of superpixels aim to over-segment the image to preserve image structure, non-overlapping features [296–304]. With many papers utilising simple linear iterative clustering(SLIC) [305] in order to achieve over segmentation [296, 298–301, 303, 304] or extensions such as Hyper-manifold SLIC [296]. Each segmentation must then be converted to a graph representation, nodes and edges. For each node we can summarise all pixels within each superpixel by simply taking the spectral mean [300–304]. Alternatively each node feature vector can be generated by another model. For example, Diao et al. utilise CNN feature maps in conjunction with spectral and locational encoding(superpixel position within the image) [301]. Alternatively, spectral transformers and 1D-CNNs can also generate learnt node feature vectors [298, 300]. Edges, much like the node information, can be established based on the use case. There are two main methodologies, spatial and spectral attention. For spectral attention, the edges are constructed between the most similar superpixels [296, 303]. Spatial edges are constructed between adjacent nodes [298, 299, 301, 304]. With some use cases utilising both

spatial and spectral edge connections [300, 302]. The number of edges is either predetermined, such as adjacent superpixels, or alternatively, the K most similar nodes establish edges, where K is a chosen parameter.

Most models that utilise either graph attention or graph convolutional networks remain fairly shallow. The choice of three layers is common in many applications [301]. As each layer of the network aggregates information from each node where an edge connection is present, subsequent layers will inherently aggregate information from 2nd order connections. This first order of second order aggregation is also known as the k hops. Limiting the number of k hops reduces the information aggregated to, in most cases, the most relevant information, as set in graph generation by the edges. Some implementations do utilise many more layers, such as encoder and decoder architectures of GNN layers [298, 299]. These models implement expansive and contrastive architectures. The hidden feature vector of each node is squeezed by the contrastive model and reconstructed by the expansive model [299]. Alternatively, each layer can also reduce or expand the number of nodes within the graph [298]. Combining two nodes is achieved via graph pooling operations and unpooling in reverse. These autoencoder-based models add a self-supervised objective, a reconstruction loss, adding additional stability to the training and forcing latent embeddings to retain only sufficient information for reconstruction and classification. However, autoencoder-based models suffer from slow convergence between the reconstruction loss and the classification task. Additionally, the architecture increases the computational complexity, requiring more time and memory for training and inference.

In this thesis, we adopt a simpler and more computationally efficient GNN design that preserves the benefits of contextual message passing while avoiding the high overhead of deep or autoencoder-based architectures. This choice supports scalability to large satellite datasets and aligns with the practical constraints of interactive labelling applications. While superpixel and GNN approaches have been explored in remote sensing, the majority of the existing work explored is formulated around supervised classification tasks. The works mentioned typically assume the availability of ground truth annotations, therefore typically evaluate performance primarily through classification accuracy. Consequently, these representations are not designed or tested for interactive dataset construction. In contrast, the graph-based representation proposed in this chapter is not used as a prediction model but rather as an intermediate representation for unsupervised similarity assessment and segment-level label extraction.

6.2 Problem Formulation

In this chapter, we address the labelling of remote sensing images at the superpixel level, utilising the same interactive two-dimensional plot.

Let

$$X = \{x_i \in \mathbb{R}^{H\times W\times B} | i = 1,...,N\}$$

denote a dataset of N remote sensing chips of spatial dimensions $H \times W$ and B spectral bands. The goal is to derive a superpixel representation for each chip

$$Y_i = \{y_{ij} \in \mathbb{R}^2 | i = 1,...,N, j = 1,...,M_i\}$$

where M_i is the number of superpixels in chip x_i, such that superpixels with similar characteristics are mapped closely together.

We consider two models to extract and encode features. A CNN feature extractor

$$f_\theta : \mathbb{R}^{H\times W\times B} \to \mathbb{R}^{H\times W\times D}$$

maps each chip, x_i, to a latent feature map $F_i = f_\theta(x_i)$. Let F_i^{L-1} denote the penultimate layer with the same height and width as the input. In our implementation, it is the penultimate layer as it has the same width and height $W \times H$, therefore requiring no interpolation for graph construction. The feature vector for a superpixel s_{ij} is then the average over all pixels in F_i:

$$z_{ij}^{CNN} = \frac{1}{|P_{ij}|} \sum_{(u,v)\in P_{ij}} F_i^{(L-1)}(u,v)$$

where P_{ij} is a set of corresponding pixel coordinates in superpixel s_{ij}.

The graph representation $G_i = (V_i, E_i), \quad V_i = \{v_{ij} \mid j = 1,\ldots,M_i\}$, is a set of nodes corresponding to superpixels $v_{ij} \in V_i$ and k number of edges $e_{jk} \in E_i$ connection neighbouring superpixels. A GNN with parameters ϕ maps node features to embeddings

$$h_{ij}^{(2)} = \Phi_\phi^{(2)}(z_{ij}^{CNN}, \{z_{ik}^{CNN} : k \in \mathcal{N}(j)\})$$

where $\mathcal{N}(j)$ denotes the set of neighbours of a superpixel s_{ij}. The second layer output $h_{ij}^{(2)}$ serves as the final superpixel encodings. The ultimate layer of h is for the loss function.

Finally embeddings are calculated from the GNN encodings

$$y_{ij} = g(h_{ij}^{(2)}), g : \mathbb{R}^d \to \mathbb{R}^2$$

Where g is a dimensionality reduction algorithm.

Loss Function

Each pixel is assigned a fuzzy membership vector

$$y_{i,u,v}^{CNN\_target} \in [0,1]^C$$

which is the target loss function for the CNN derived from fuzzy clustering. Where u and v are the positional encodings defining a pixel in chip i. Therefore minimising the loss between $\mathscr{L}(z_i, y_i^{CNN\_target})$ for each chip.

Likewise, we can average the fuzzy membership to represent each superpixel

$$y_i^{graph\_target} = \frac{1}{|P_{ij}|} \sum_{(u,v)\in P_{ij}} y_i^{CNN\_target}(u,v)$$

is used as a loss function to train the GNN, $\mathscr{L}(h_{ij}, y_{ij}^{graph\_target})$.

Key assumptions:

- Local neighbourhood in the latent encoding space $h(2)_{ij}$ is semantically similar to other superpixels.
- Bipartite graph matching between superpixels in x_i removes the dominant textural orientation problem, as seen in the previous chapter.
- The generated superpixels are appropriately constructed for user interpretation and labelling.
- The CNNs textural features is enough context for the GNN to be able to encode spatial information within each superpixel.
- Fuzzy clustering provides a suitable generalisation of land-use and ocean classes.

6.3 Motivations

Within an RS image each pixel holds significance for labelling, especially when considering small spatial resolutions. Given the size of RS datasets, considering each individual pixel for any algorithm increases the computational costs. The increase in complexity holds true for supervised approaches that are only segmenting pre-learned features, and more so for generalised and unsupervised approaches that model all potential discriminating factors of a given pixel.

However, given the natural pattern of adjacent homogeneous pixels present within remote sensing images, we can exploit superpixels to reduce this complexity while preserving important structure. Superpixels allow for small parcels of similar adjacent pixels to be represented as a single pixel via summarised statistics of the group.

RS images also, in general, follow the first rule of geography: things that are closer are more related than those that are further away. The relation can be modelled as a graph, which can be seen in the literature of the background section, where the edges of a graph are connections between two related, close features. Edge connections are often implicit in their local neighbourhood connections or explicit. Explicit connections are formed by forming edge connections between two nodes, or superpixels, that are adjacent. As most works focus on a smaller chip of the original image, usually $\leq 256 \times 256$ pixels, any edges formed within the chip implicitly act as localised neighbourhood connections, even when edges are formed between the most similar superpixels rather than strictly adjacent ones. Edges within a graph network do not encode any information about the cardinal directionality of connected nodes. Only weightings that are assigned to edges, if any, are the strength of the connections. Therefore, potentially removing any bias towards the orientation of textures as we have seen in the previous chapter.

The most important part of modelling superpixels as a graph network is in the assignment of information within the node, the superpixel's summarised statistics of the whole group. Low-cost approaches can simply utilise the averaged values of the raw data, the spectral values in each band, for each node. However, averages do not encode any significant textural features, which can be crucial. For this reason, some methods implement a CNN model to incorporate optimal textural information, with graph neural networks encoding spatial information across superpixels. In this chapter, we approach the modelling of RS images in this way with a CNN feature extractor for nodes and a GNN for spatial encodings.

A U-net is utilised as a textural feature extractor for two reasons: it is efficient at classification and also retains an activation map of the original width and height of the original image. Having the same size as the original image allows for a one-to-one mapping in order to get a superpixel representation without interpolation. This is important because our superpixel features are computed directly by averaging pixel-level CNN activations, therefore, any mismatch in spatial resolution would require additional upsampling operations that may introduce artefacts or degrade feature consistency. Other models can achieve the same result via transposed convolutions, however, U-net already achieves this and is commonly utilised in

literature. Other standard classification backbones(e.g. ResNet) was not adopted, as they are more useful for semantic abstraction through consecutive spatial pooling, whereas this work requires dense, pixel-aligned feature maps. We utilise fuzzy c-means as a target in the loss function in order to keep as much generalisability within the activation maps. This also requires tests to see which number of fuzzy clusters is most appropriate for our solution.

Verification of embedding spaces also poses a significant problem for labelling systems. Without pre-existing labelled samples, a human user is needed for qualitative analysis, which is costly. Therefore, we also proposed a graph segmentation analysis using a pre-existing metric to compare neighbourhoods of local node features. In this way, we can quantise the differences between two latent spaces.

These are subsequent aims for the chapter:

- A1: Develop a granular representation of remote sensing chips that supports interactive labelling
- A2: Encode both textural and spatial contextual information for each superpixel to create semantically meaningful segment embeddings.
- A3: Construct a 2D embedding space where similar superpixels from different chips are positioned closely together, enabling cross-image label propagation.
- A4: Evaluate the stability and interpretability of the embedding space without requiring labelled data.
- A5: Demonstrate that superpixel representations improve the expressiveness and usability of the interactive labelling tool compared to chip representations.

These are the hypotheses:

- H1: Superpixel-based feature embeddings produce a more structured latent space for fine-grained remote sensing analysis than scene-level representations alone.
- H2: Encoding spatial adjacency through graph-based representations preserves local geographic context within learned superpixel embeddings.
- H3: Superpixel embeddings group segments that are semantically meaningful to human interpretation, even in the absence of explicit supervision.

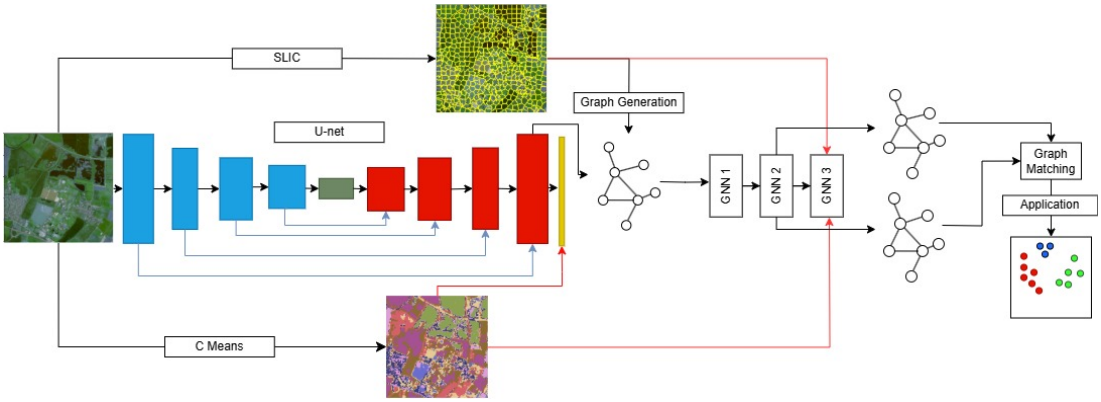


Figure 6.1: Shows the information flow of data within the pipeline. Blue boxes denote the constricting layers of the U-Net with red denoting the expansive layers. Yellow being the predictive layer. Blue lines indicate a skip connection. Red lines indicate the use of data within the loss function.

- H4: Superpixel embeddings extracted from independent image chips form a shared latent space in which similar geographic features can be explored and grouped across scenes.

6.4 Methodology

The proposed pipeline contains three key elements (Fig. 6.1) – a U-net for feature extraction; a Graph Neural Network (GNN) for extracting relational embeddings; and a graph-matching algorithm to find the similarity between two graphs. For information on the dataset utilised please refer to the previous chapter 5 section5.4.1. Within this work we utilise a singular tile for visualisation due to constraints on the matching algorithm and processing time. The application interface and implementation are identical to the previous chapter 5, apart from small visual changes. The only major change is the interface functionality for labelling segmentations. The user is initially displayed chip-level segments and can choose to branch or project a subset of the data for segmentation labelling or chip-level labelling. Segments are displayed as points on a two-dimensional plane with the same selection interaction, the display highlights the chosen segments from the images by adjusting the brightness of the unselected segments. The choice for initially presenting chip-level images is two-fold: to reduce the computational complexity of projecting a large quantity of points (≈ 500 per chip) and for the user to easily find features of interest at the scene level before addressing fine-grained labelling.

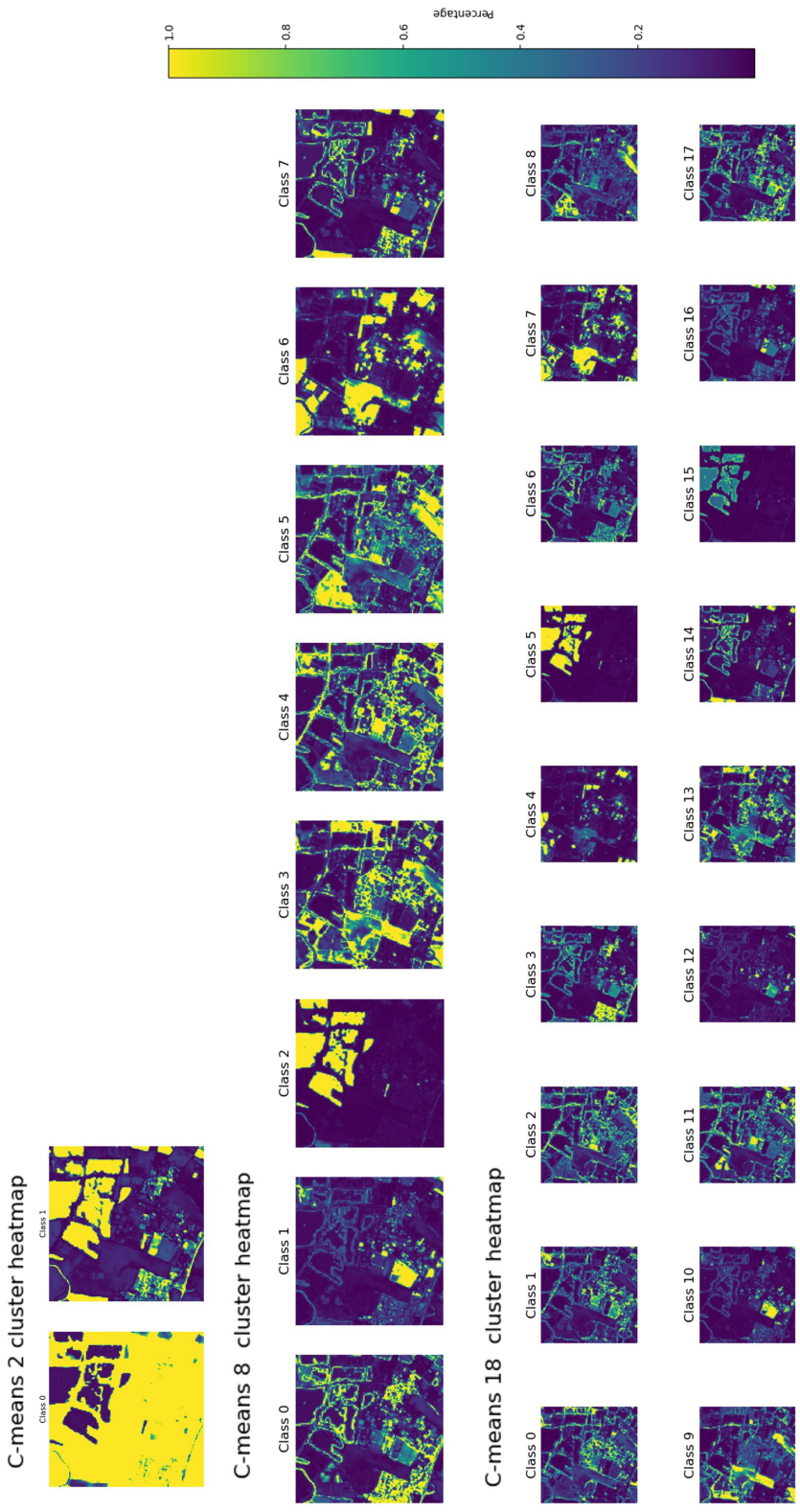


Figure 6.2: Cluster separation and percentage association for different C-means. Demonstrated on a singular chip from tile T30UYC.

6.4.1 Fuzzy C-Means target extraction

Due to the spatial resolution of most satellite platforms, pixel assignment can be ambiguous or uncertain [306]. Fuzzy C-means extraction, unlike its K-means counterpart, assigns to each data point a degree of membership to each cluster. In this work, the number of clusters was treated as a variable, and values of $C = 2, 8, 18$ were evaluated. All 12 spectral channels were used as input features. To reduce memory consumption and computational cost during model training, pixel-level subsampling was applied. Specifically, every 56^{th} pixel was sampled from the image when forming the training set. This sampling procedure was motivated by the clustering algorithm's memory constraints. Additionally, a stride was chosen that was not divisible by the width of the image 1098 to enforce that chosen pixels were not adjacent, increasing variation.

See figure 6.2, for an example of the clusters produced when applying this method. Cluster association in this way introduces some variability to each pixels association to a particular cluster. This is similar to endmember extraction in which we do not assume that a reflected pixel is entirely composed of one material; there are multiple abundances for each pixel. However, C-means extracted centroids are calculated from the quantity of pixels and density, whilst endmembers are calculated as "pure pixels". Therefore, each cluster in C-means does not necessarily map to a particular material, such as water, concrete or vegetation, but can imitate such extraction.

6.4.2 Pre-Proccessing

For processing in 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 chip size is a balance between training size and the preservation of contextual neighbourhood information. The larger the image, the more computationally expensive the resulting pipeline becomes; however, more information about neighbouring features can be encoded.

6.4.3 CNN feature extraction

To learn the texture representation we utilise a CNN where the learning function considers the C-Means classification for an objective discrimination of features. The CNN of choice was a U-net, with 5 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 vice versa for the decoder. The model's target is the fuzzy C-Means predictions. We used 2, 8, and 18 clusters to evaluate 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 [14].

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

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

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 for each class and average. The final loss combining the two is a variation of combo loss [307].

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.

6.4.4 Graph Construction

Nodes for the graphs are determined by the Simple Linear Iterative Clustering (SLIC) [305] algorithm, where each segment is considered a new node. We set the target number of segmentations for SLIC as $N = 500$. N is only a target, as SLIC can vary N based on each sample, as it splits or merges segments. 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 the K geographically nearest segments. We tested on eight K nearest neighbours. This work differs to previous work by [308] because we only include information from the CNN and rely on geographical neighbours to construct edges, removing any mean spectral

information in the node. Whilst their existing works aim to achieve classification on a dataset, this work is focused on semantic embeddings of segments for exploration and labelling. In addition, we utilise fuzzy clustering as a targeted loss, which is derived from spectral information. Therefore, introducing spectral information explicitly in each node may lead the model to simply learn the mapping produced by fuzzy clustering. The resulting representation is a Graph $G(V,E)$ with $N = V$ number of nodes with 64 features F each. There are $K \times N$ unweighted edge connections between nodes, $\approx 8 \times 500$ edges.

6.4.5 Graph Neural Network

The graph attention network is three layers with attention mechanisms [309]. 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, X'_i can be calculated with the following:

$$X'_i = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in N(i) \cup i} \alpha_{ij} W^k x_j\right)$$

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

$$\alpha_{ij} = \frac{\exp\left(a^T \text{LeakyReLU}\left([Wx_i \| Wx_j \| e_{ij}]\right)\right)}{\sum_{k \in N(i) \cup i} \exp\left(a^T \text{LeakyReLU}\left([Wx_i \| Wh_k \| e_{ik}]\right)\right)}$$

where 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 visualise 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 outputs of the second-to-last GAT layer; within this layer, we constrain the feature output to 8 hidden feature vectors with 8 attention heads. Therefore, our final representation is a 64-dimensional vector.

Similarly, we tested a 3-layer graph convolutional network (GCN) with an embedding vector of 60 hidden features. The choice to test both networks is to determine whether the inclusion of an attention mechanism affects the information at each node relative to its neighbours.

6.4.6 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 and edges the connection to the nearest nodes in feature space, X, from the second layer of the GNN. Graph matching finds how similar two graphs are based on their node and edge composition. As our approach utilised many segments our resulting nodes are in the hundreds, therefore, utilising multi-graph solvers which take into account both vertices and edges is infeasible in both memory and computational complexity. The approach relies on the GNN to encode the edge relationships. We simplify the matching to a bi-partite graph representation using the Hungarian algorithm [310] to match nodes present in both graphs and calculate the overall similarity based on the Euclidean distance between matched nodes in feature space X. Applying the matching to create a similarity matrix for every chip processed through UMAP [311], creates a two-dimensional embedding for visualisation. This bipartite matching is the most computationally demanding component of our pipeline. Hungarian matching has $O(N^3)$ computational complexity and for our implementation N is 500, dependent on SLIC. To mitigate runtime cost, the graph matching is computed offline and implemented using a GPU and parallel processing.

6.4.7 Evaluating Schema

We use a qualitative approach to evaluate this method, using an interactive application that allows the user to explore and label the final features produced in two dimensions via UMAP for exploration and labelling. To embed chip images represented by a graph structure we need to define the distance between two graph structures. We also evaluated the CNN component of the architecture for accuracy utilising a pre-existing dataset.

6.4.8 Graph Evaluation

We evaluated the segmentation feature space, X, produced by each GNN. We compared each segment to its closest segment within the feature space. The comparison was made using four common similarity measures: gray-level co-occurrence matrix (GLCM) [312], local binary patterns (LBP) [313], Structural similarity index measure (SSIM) [314], and spectral angle mapper (SAM) [315]. Next four paragraphs added to give background information on the similarity measures. A further fifth paragraph provided motivation for the selection. GLCM characterises second-order texture statistics by modelling joint pixel intensity occurrences at

fixed spatial offsets. GLCM-based features such as contrast, homogeneity and correlation have been used in remote sensing for texture or crop classification [316]. Even with modern methods, GLCM-based features remain competitive when combined with spatial/spectral information [317]. The statistics derived from GLCM capture local pixel patterns; however, the algorithm is sensitive to window size, sampling directions, and does not model more complex spatial relationships.

LBP is a popular method for representing local spatial texture due to its simplicity, low computational cost and relative robustness to grayscale changes. In recent remote sensing and hyperspectral classification, LBP spatial features have been used to support spectral features in semi-supervised frameworks [318]. However LBP descriptors are not well suited for small features or irregular regions, which is important to note when utilising it for segments rather than small patches.

SSIM was originally developed for perceptual image quality assessment [314], comparing regions based on summarised statistics. Three main statistics are extracted: luminance, contrast and structure. SSIM has been adopted in remote sensing to assess radiometric consistency or image reconstruction fidelity [319, 320]. Whilst SSIM can gauge the similarity in intensity distribution and contrast between images, it lacks the ability to model textural differences efficiently and is highly dependent on the absolute mean and standard deviation cite 7025115. Pambrun et al. also note SSIM is subject to distortion near hard edges and the measure should be avoided where statistics are not fairly homogenous [321].

Spectral angle mapper is a physics-based metric to ascertain the similarity between two spectra. As remote sensing images, multi- and hyper-spectral, images are composed of spectral information the metric is widely used [322, 323]. SAM approaches similarity by considering each pixels as a vector and calculating the angle between them, which can be interpreted as assessing material similarity, see section 2.6. The metric is normally calculated per pixel however we demonstrate its effectiveness on superpixels, given the assumption that they are homogenous.

The selection of algorithms was to strike a balance between textural and spectral similarity. Spectral similarity will aid in identifying segments with similar materials according to the intensities of each wavelength returned. Whilst texture is less suitable for segmentation analysis, as most segments by design are homogenous selections of material. We introduce it to include some neighbourhood pixel information, using the minimum encompassing bounding box, and secondly, to differentiate very similar spectral returns based on texture. For example,

industrial buildings can be very similar to urban buildings; however, they can be differentiated by their size. Therefore, we must consider a balance of measures. The first two measures were calculated using a bounding box that enclosed the segment to overcome their limitations when applied to smaller segmentations, while the latter two were based solely on the content within the segment.

In addition, we conducted a second amended test including the local neighbourhood encoding of each model. Each segment comparison also included the nearest eight geographical segments. By utilising graph matching, we can apply each metric to the 8 neighbouring segment pairs and average for a similarity measure.

Figure 6.3: Example of segments within a chip taken from the testing data. The highlighted segments are utilised for comparison in this section.

This section presents the methodology and reasoning behind the tests performed. We first cover the similarity measures. Then the segmentation similarity, and the inclusion of neighbourhood similarity.

Gray Level Occurrence Matrix

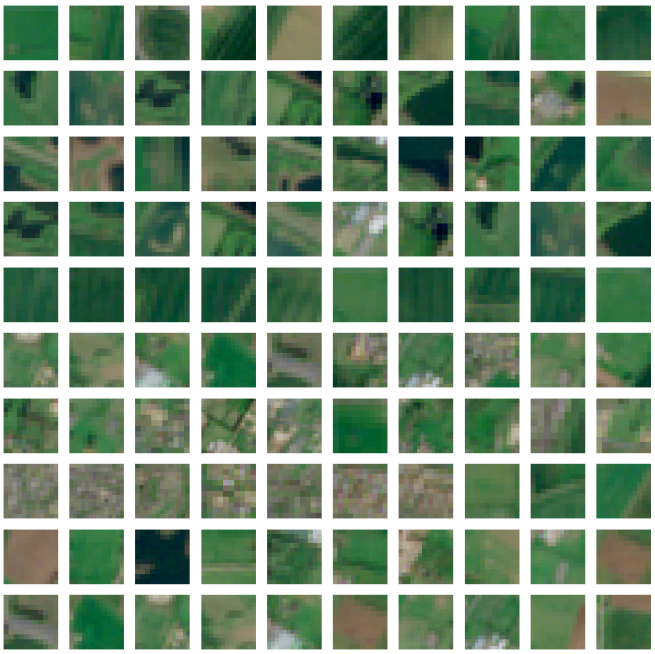


Figure 6.4: Each row shows the most similar segments to the first picture in each column according to GLCM. The segments are ordered from north to south as shown in 6.3.

As the name suggests, the GLCM will calculate how often a pixel, i, will share the same intensity as another pixel, j, within the image. The comparison of pixels j from i is only computed for given directions. For example, horizontal, vertical and diagonal lines can be traversed from pixel i for each pixel j visited on these lines, and the intensity is recorded. In this fashion, a GLCM matrix can be created. Where values in the matrix, positioned at $GLCM_{matrix}[i, j]$, indicate how often a grey value pixel i and grey value pixel j occur along the given directions. After a matrix has been made, four statistics can be derived: contrast, correlation, energy and homogeneity [312]. Contrast measures local variation. Correlation measures the joint probability of occurrence of specified pixel pairs. Energy correlates to the uniformity of the image. Lastly, homogeneity, is a measure of smoothness, how consistent are pixel pairs.

We compute the bounding boxes for each segment before calculating similarity, as GLCM is unstable over small regions. We take the four horizontal and vertical directions from each pixel i and sample along them for each pixel, and then for every other pixel. This can be expanded to diagonal lines and different sampling rates. Given the minimal area coverage of

each segment, we maintain small sampling rates. Figure 6.4 shows the most similar segments to the first segment in each row, from a singular chip. The similarity between the first segment and the other segments in each row is primarily based on texture and not the spectral value. For example, barren fields, perceptual brown segments are considered similar to vegetative fields that have a similar uniform texture.

Local Binary Patterns

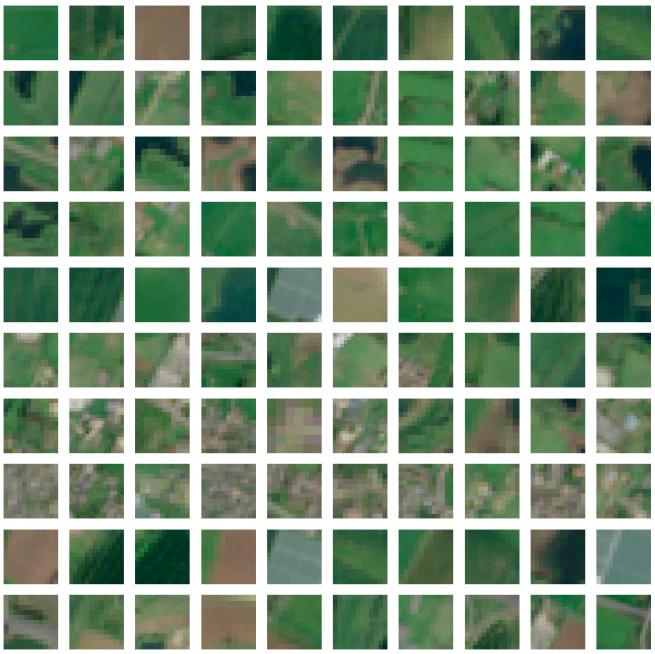


Figure 6.5: Each row shows the most similar segments to the the first picture in each column according to LBP. The segments are ordered from north to south as shown in 6.3.

Local binary patterns create a histogram based on the intensity of neighbouring pixels. Choosing a central pixel, i, the local neighbourhood can be taken as those in a circular region around i. For each pixel j in the neighbourhood, assign a binary value dependent on the threshold, pixel i. As this is based on intensity, the images must be converted to greyscale beforehand. The resulting binary pattern can be converted into a decimal. Once repeated for each pixel, a histogram can be computed for comparison; we then calculated the cosine distance between the histograms. Within our implementation, we define the neighbourhood for pixel i as an 8-pixel radius circle. Similar to GLCM, the segments are computed with the smallest bounding box. As the measure is based on texture, smaller and non-rectangular segments would negatively impact the measure. Inherently, this includes some neighbouring pixels, regardless of whether a minimal bounding box was selected. Figure 6.5 shows examples of the most similar segments according to LBP within a singular chip, where each row is based on the most similar segments to the first segment in that row. As shown, LBP similarity is more consistent with texture, much like the previous example from GLCM. However, unlike GLCM, LBP is more capable of measuring textural similarity between urban areas but suffers

in comparison for vegetative samples.

Structural Similarity Measure

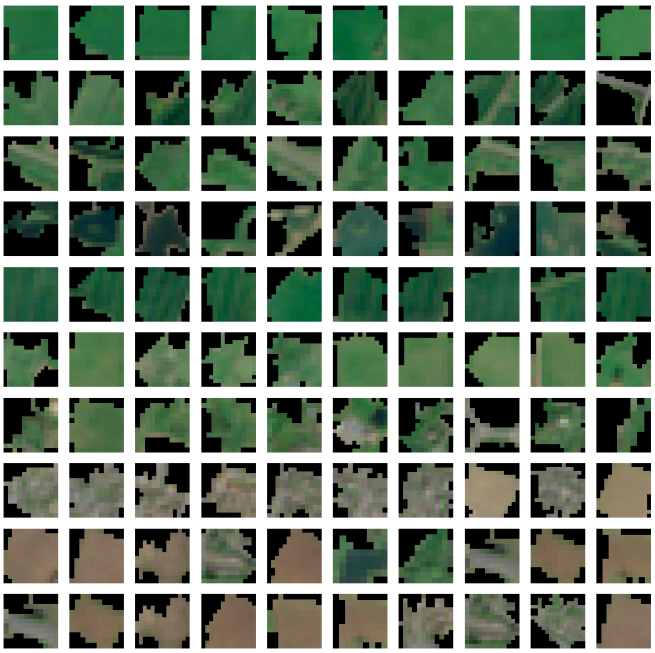


Figure 6.6: Each row shows the most similar segments to the first picture in each column according to SSIM. The segments are ordered from north to south as shown in 6.3.

Given we have compared the textural differences between each image, we can now compare the multispectral variations between each segment. SSIM is comprised of three measures: luminance, contrast, and structure. Luminance is a measure of the average brightness, and contrast is the fluctuations between intensities. We don't calculate the structure within our measure as we compute the GLCM and LBP for pattern and textural analysis. The algorithm utilises the mean, μ, and standard deviation, σ, of each image.

Luminance is calculated with:

$$L = \frac{2\mu_x\mu_y + C_1}{\mu_x + \mu_y + C_1}$$

Where x and y are the two segments being compared. C_1 is a small constant for stability.

Contrast likewise is:

$$C = \frac{2\sigma_x\sigma_y + C_1}{\sigma_x + \sigma_y + C_1}$$

As this method only utilises the mean and variance, we can utilise just the information present within the segment, removing the need for bounding boxes. The final measure can be

taken as $SSIM = C * L$, given that the structure measure is excluded. Figure 6.6 shows the most similar segments in each row to the first sample in the row, taken from a singular chip. SSIM as expected produces very perceptually similar segments, however, it can be seen to convolute similarity between barren land and urban areas. However, SSIM excels at vegetative segments where similar spectral returns are correctly identified.

Spectral Angle Mapper

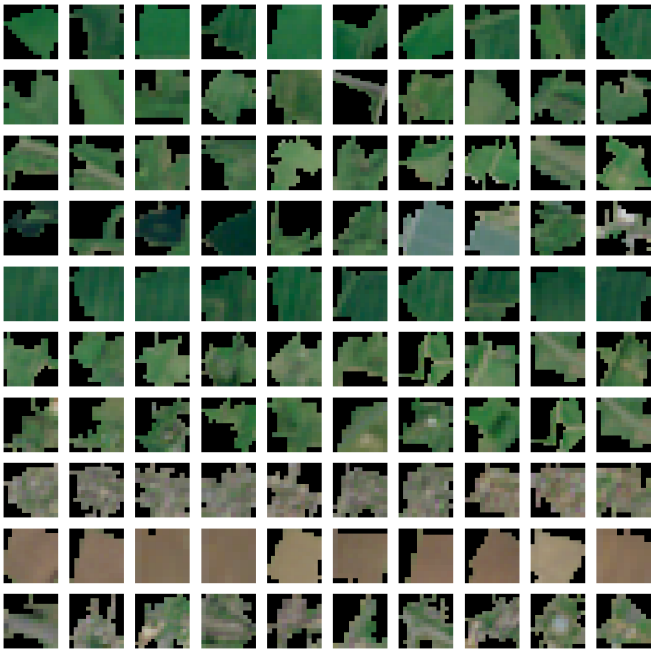


Figure 6.7: Each row shows the most similar segments according to SAM. The segments are ordered from north to south as shown in 6.3.

Lastly, we utilise SAM for its simplicity and ability to compare spectral signatures. Intuitively, the similarity is calculated by treating each pixel as a vector and calculating the angle between them. Since a segment contains multiple pixels, we calculate the mean of each wavelength to represent the segment. Since each segment should, in theory, be homogeneous, the noise introduced by taking an average is minimal. As shown in figure 6.7, where each row contains the most similar segment to the first segment in the row, SAM is a superior measure of similarity. However, certain segments of vegetative land are considered similar to industrial roofing, see row 4 where the roofing is the perceptually light grey segments.

Summary Across the evaluated similarity measures, clear differences emerge between the efficiency of each algorithm with respect to textural and spectral properties. Texture-based

methods, GLCM and LBP, group segments based on spatial uniformity and structural pattern rather than spectral properties. Both methods are effective at identifying regions with similar roughness or land use cover. However, where GLCM suffers to extract textural similarity, LBP excels, and the same can be said for LBP to GLCM. GLCM shows sensitivity to spectrally distinct but texturally uniform regions, such as barren lands and agricultural fields. LBP demonstrated improved discrimination within urban areas but performs less reliably for vegetative areas where repetitive patterns are limited.

The spectral similarity measures , SSIM and SAM, exhibit consistent grouping of segments based on perceptual and spectral similarity. SSIM shows consistent similarity between vegetative segments, however, fails to capture the difference between urban and barren regions. SAM shows the most consistent similarity between segments with regards to both spectrally similar and semantically similar segments. However, as SAM ignores textural differences, there are some "outliers" within the displayed result, notably large industrial segments with uniform reflectance values.

These results demonstrate that no single measure captures all relevant aspects of remote sensing segments for similarity comparison. Their complementary behaviour supports the use of multiple similarities as a criterion for testing if the learned feature space, X, corresponds to meaningful encodings, both texturally and spectrally.

Graph Test 1

In Algorithm 1, we provide an outline of the segmentation comparison algorithm. This test provides a simple basis for checking the nearest segment for similarity in a hidden feature space we refer to as X, where X is analogous to H. The graphs utilised are the same as those generated during the training and testing process detailed in Section 6.4.4. Each graph is fed forward in the GCN, and the features, h, are extracted. We now construct a KD tree to efficiently search through the high-dimensional set of features, H. Any method to find the closest segment can be utilised. For each feature, we find its nearest neighbour. This pairing is not always symmetrical. Both bounding boxes are computed, and the similarity metrics, as mentioned earlier, are calculated.

Graph Test 2

The second test constructed is an expansion of the first, see Figure 6.8, introducing the concept of geographical neighbours. Given that images and, more broadly, RS images are two-dimensional, with an x or y axis representing longitude and latitude, there exists a geographical neighbour to any segment. For this test, we want to compare the similarity between

Algorithm 1: Graph Segmentation Analysis

Segmentation_Similarity($Gnn, Graph, Segments, Images$)
for all $g_{i..n} \in Graph$ **do**
  $h_i \leftarrow Gnn(g_i)$
end for
Build KDtree $KD(H)$ where each node is $h \in H$
$total_{glcm} \leftarrow 0$
$total_{lbp} \leftarrow 0$
$total_{ssim} \leftarrow 0$
$total_{sam} \leftarrow 0$
for all $h_{i..n} \in H$ for all i to n **do**
  $h_j \leftarrow$ second result in $KDQuery(h_i, k = 2)$
  $s_i \leftarrow Segment_i$
  $s_j \leftarrow Segment_j$
  $b_1 \leftarrow computeBoundingBox(Image, s_i)$ corresponding to h_i
  $b_2 \leftarrow computeBoundingBox(Image, s_j)$ corresponding to h_j
  $b_1 \leftarrow Greyscale(b_1)$
  $b_2 \leftarrow Greyscale(b_2)$
  $dist_{glcm} \leftarrow GLCM(b_1, b_2)$
  $dist_{lbp} \leftarrow LBP(b_1, b_2)$
  $dist_{ssim} \leftarrow SSIM(s_1, s_2)$
  $dist_{sam} \leftarrow SAM(s_1, s_2)$
  $total_{glcm} \leftarrow total_{glcm} + dist_{glcm}$
  $total_{lbp} \leftarrow total_{lbp} + dist_{lbp}$
  $total_{ssim} \leftarrow total_{ssim} + dist_{ssim}$
  $total_{sam} \leftarrow total_{sam} + dist_{sam}$
end for
$average_{glcm} \leftarrow total_{glcm} \div n$
$average_{lbp} \leftarrow total_{lbp} \div n$
$average_{ssim} \leftarrow total_{ssim} \div n$
$average_{sam} \leftarrow total_{sam} \div n$

Algorithm 2: Graph Segmentation Neighbourhood Analysis

Segmentation_Similarity($Gnn, Graph, Segments, Images$)
for all $g_{i..n} \in Graph$ **do**
  $h_i \leftarrow Gnn(g_i)$
end for
Build KDtree $KD(H)$ where each node is $h \in H$
$total_{glcm} \leftarrow 0$
$total_{lbp} \leftarrow 0$
$total_{ssim} \leftarrow 0$
$total_{sam} \leftarrow 0$
for all $h_{i..n} \in H$ for all i to n **do**
  $h_j \leftarrow$ second result in $KDQuery(h_i, k = 2)$
  $s_i \leftarrow Segment_i$
  $s_j \leftarrow Segment_j$
  $Geo_i \leftarrow GeographicalNeighboursH(s_i)$
  $Geo_j \leftarrow GeogrphicalNeighboursH(s_j)$
  $P = HungarianMatching(Geo_i, Geo_j)$
  $pairs_{glcm} \leftarrow 0$
  $pairs_{lbp} \leftarrow 0$
  $pairs_{ssim} \leftarrow 0$
  $pairs_{sam} \leftarrow 0$
  for all $(H_{ij}, H_{ji}) \in P$ **do**
    $b_1 \leftarrow computeBoundingBox(Image, s_{ij})$ corresponding to h_{ij}
    $b_2 \leftarrow computeBoundingBox(Image, s_{ji})$ corresponding to h_{ji}
    $b_1 \leftarrow Greyscale(b_1)$
    $b_2 \leftarrow Greyscale(b_2)$
    $pairs_{glcm} \leftarrow pairs_{glcm} + GLCM(b_1, b_2)$
    $pairs_{lbp} \leftarrow pairs_{lbp} + LBP(b_1, b_2)$
    $pairs_{ssim} \leftarrow pairs_{ssim} + SSIM(s_1, s_2)$
    $pairs_{sam} \leftarrow pairs_{sam} + SAM(s_1, s_2)$
  end for
  $total_{glcm} \leftarrow total_{glcm} + (dist_{glcm} \div |P|)$
  $total_{lbp} \leftarrow total_{lbp} + (dist_{lbp} \div |P|)$
  $total_{ssim} \leftarrow total_{ssim} + (dist_{ssim} \div |P|)$
  $total_{sam} \leftarrow total_{sam} + (dist_{sam} \div |P|)$
end for
$average_{glcm} \leftarrow total_{glcm} \div |H|$
$average_{lbp} \leftarrow total_{lbp} \div |H|$
$average_{ssim} \leftarrow total_{ssim} \div |H|$
$average_{sam} \leftarrow total_{sam} \div |H|$

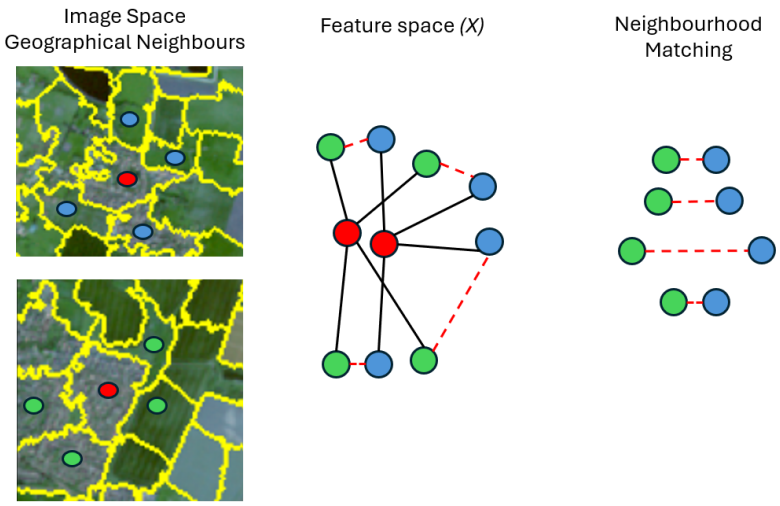


Figure 6.8: This figure shows a simplified explanation of our second graph test. The segments have not been over-segmented, nor do they reflect the 8 neighbours used in this work; however, they show 4 for ease of understanding and to demonstrate the core algorithm. The two red circles convey a potential pair of nodes that are similar in feature space, X, due to their similar content. The four blue and green circles are the geographical neighbouring segments of each respective node, denoted by solid black lines, in image space and feature space. The red lines represent the distance, shown by the length of the line, in feature space, X, between each neighbouring node. Graph matching calculates which nodes to pair, red lines, by bi-partite matching so that the overall distance is minimal as reflected in feature space, X. The metrics(SAM,LBP,SSIM,SAM) are then calculated on each node pairing and averaged. In this way we can assess how much information from neighbouring nodes is encoded into our feature space.

the geographical space and the feature space X that we extract from our GNNs. A quantitative method to assess the shared information during message passing. In our implementation, SLIC produces relatively uniform rectangular segmentations and therefore has roughly four to nine segmentations that share a border, or are neighbouring. Those neighbouring segmentations also have a corresponding position in feature space X. If our feature space encodes segments with similar geographical neighbours closer together, then the geographical neighbours must also be close in feature space,X. To test this, we must compare the similarity of the geographical neighbours of both segments that lie close in feature space.

In Algorithm 2, we detail how we implemented the test. Given two segments that are close in feature space, we find the resulting features of their respective geographical neighbours, denoted by $GeographicalNeighboursH$. Instead of averaging the similarities between all possible resulting geographical segments, we opt to utilise Hungarian matching to find the bi-partite pairs. These pairs are found to be the closest, given our GNN feature space. This takes each neighbouring geographical segment into account individually.Therefore, imposing

that each node must have an exactly similar neighbourhood, of spectral content, to another node close in feature space, X, which averaging would not accomplish.

6.5 Results

6.5.1 Remote Sensing Labelling Application

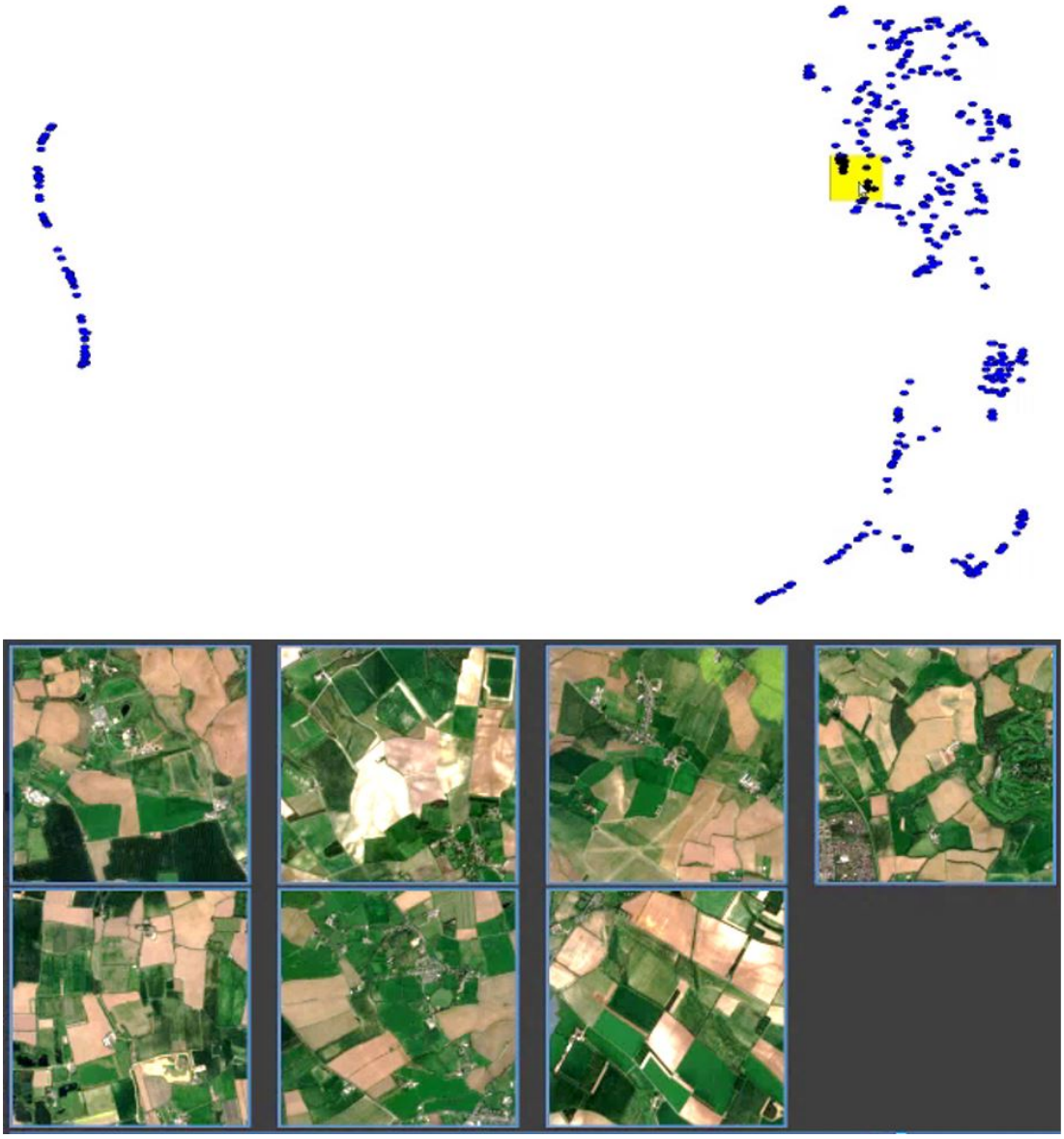


Figure 6.9: UMAP dimension reduction to 2D on the graph matching output of our entire pipeline. At this level, each point represents one chip. The user interactively highlights a resizable region which can be dragged across the manifold representation. Chip images represented by the 2D points within the highlight are displayed in the pane below.

Our pipeline enables advanced interaction within the labelling tool. Referring to the pro-

vided video of the tool utilised on the dataset, this section presents frames from the video to demonstrate and discuss functionality.

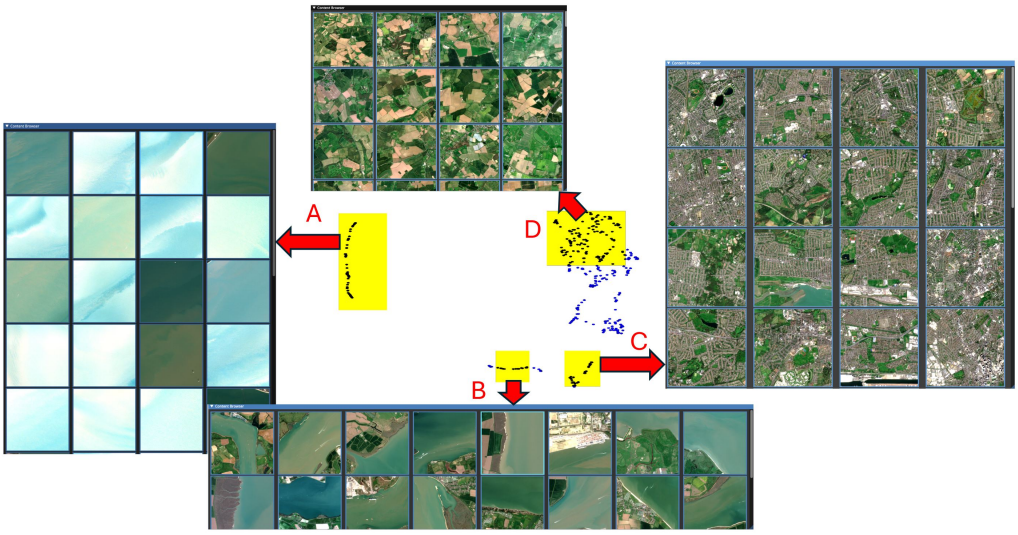


Figure 6.10: Example of the U-map embedding dimensionality reduction of high-dimensional feature space, X, output from the final graph matching stage of our entire pipeline. The images selected by the user are shown in the display pane.

The main contributions are **(6.5.2)**, the tool allows exploration of feature space, X, via a two-dimensional interface, where our pipeline has resulted in successful organisation of the high-dimensional manifold. The resultant clusters are compact and contain visually related images. **(6.5.3)** Embedded images with large texture gradients have no adverse impacts on the similarity measure. **(6.5.4)** we significantly extend previous work by enabling this interaction on segmentations within the larger images, thus allowing fast labelling at a finer granularity.

6.5.2 Cluster exploration

In figure 6.9 the user interacts with the top chip level manifold 2D representation by brushing the data points. The highlighted data points correspond to images in the source data which are then displayed in the image pane. This cluster has a patchwork of fields. There is an evolving nature to the manifold as seen previously from farmland, industrial, residential, increasing water (lakes, sea), and so on.

Figure 6.10 shows more example content of the embedding space. Samples shown from the regions contain mainly sea (A), land with water (B), dense built-up areas (C) and farmland (D). Area A in the embedding consistently has a majority water content, and 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.

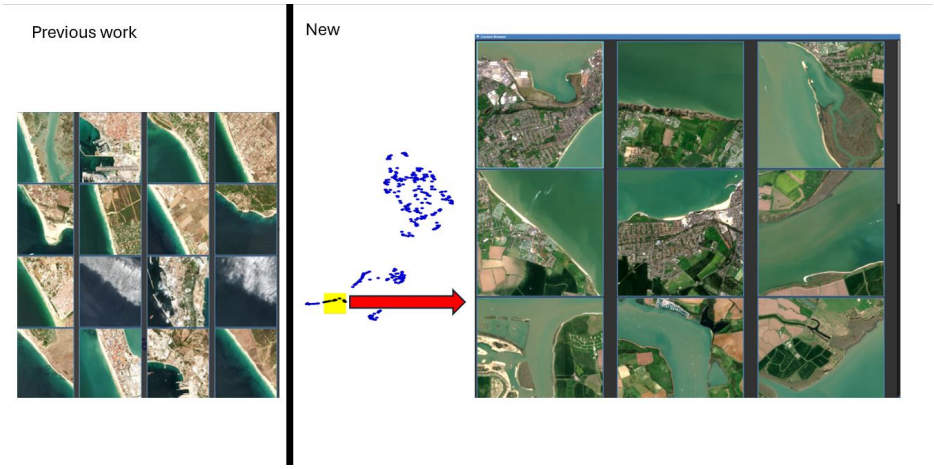


Figure 6.11: Example of the UMAP embedding space from our previous pipeline, chapter 5, and comparatively our new work. As shown the clustering in the previous work is heavily impacted by the textural orientation within a chip. Clusters in this work are now independent of orientation, there is no common textural orientation affecting cluster outcome

.

6.5.3 Rotational Invariance

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 6.11. Our previous work 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 6.10. In figure 6.11, we have selected a cluster also of approximately 50/50 land/water coverage, which indicates there is no obvious common directional gradient.

6.5.4 Segmentation Analysis

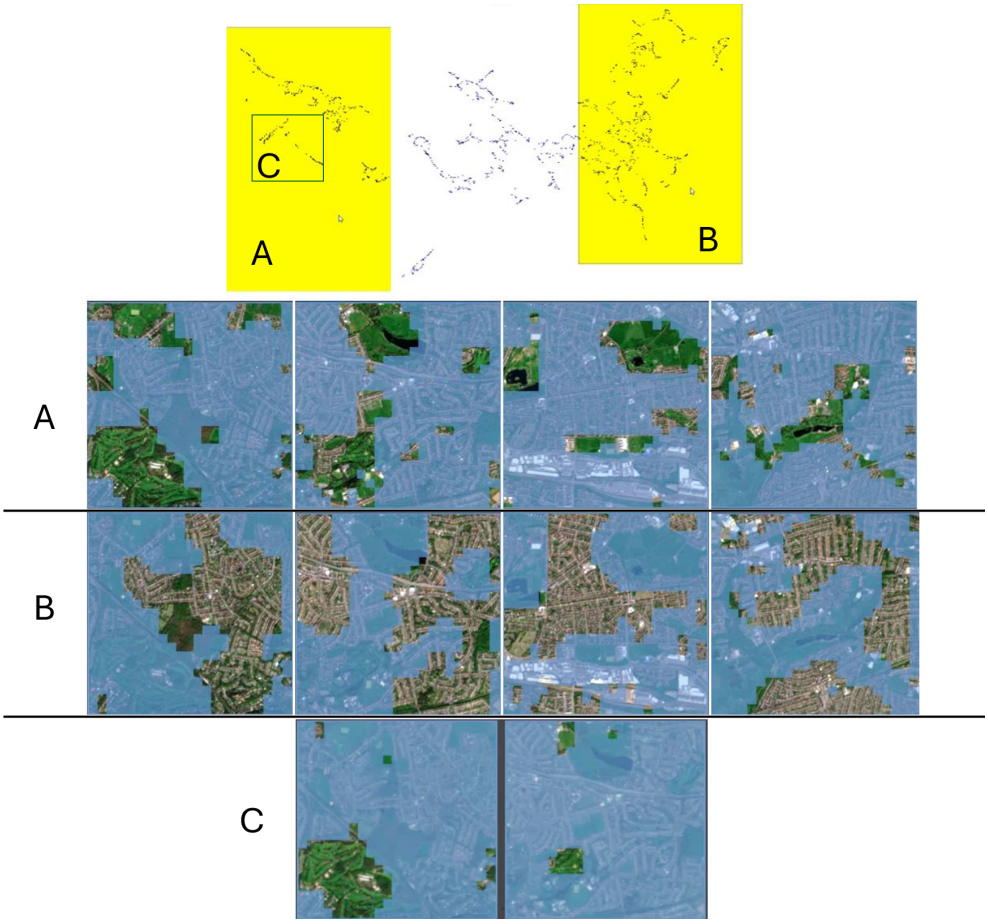


Figure 6.12: Example (from video) of projecting four images and exploring their segmentations to label as urban or vegetation. The left selection, (A), shows the largely vegetation segments in the first row of images and (B) largely urban development in the same four images, but demonstrated in the second 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).

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 2D projection of the feature space, X, is displayed for interaction, where each point represents a single (SLIC) segmentation within the images. Similar to image exploration, our approach successfully organises the manifold, with segments that are very closely related. In figure 6.12, we see at the top the 2D interactive display where each point represents one segment within the image data. By brushing multiple 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: the left selection shows vegetation with some urban features (top row of images), and the right selection shows 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 the larger and smaller golf courses irrespective of their sizes. The video demonstrates an example of how to build a labelling of the golf course data through selecting and filtering data points in the manifold.

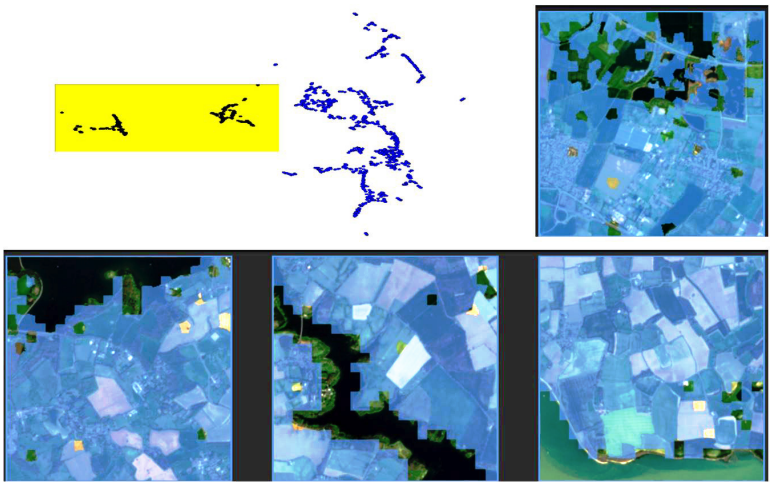


Figure 6.13: Example of extracting water features at the segment level. Various water features, including lakes, rivers and the sea. Here, as in figure 6.12, the light blue indicates the areas masked away, and the full colour segments (which are dark because they are water features) are the selected areas for labelling.

The projection at the segment level has also clustered the various water bodies within the images (figure 6.13). Via exploring the manifold, the user is able to make a selection that segments water features effectively. We found that various forms and sizes of water bodies, including large bodies of water, are easily selected via the interface and labelled simultaneously.

6.5.5 CNN Test

The following tests aim to demonstrate that our U-Net achieves near-state-of-the-art performance, thereby enabling the remainder of our GAT, graph matching, and interactive application pipeline.

Method	Accuracy
EuroSat ResNet-50	96.43
C=2 linear layers	93.67
C=8 linear layers	95.16
C=18 linear layers	92.87

Table 6.1: Respective accuracies for each model tested where C is the initial C means clustering number. Results labelled EuroSat where taken from [2].

In order to validate the utility and generalisable application of the methodology, we want to demonstrate that the trained U-net (trained using unsupervised C-means clustering) is generalisable to data not used for model training. In order to evaluate the U-net we compare to a benchmark experiment on the EuroSat dataset [2]. The dataset consists of 27,000 labelled images from the Sentinel 2 mission covering ten separate classes. We resized the images from 64x64 to 256x256 to match the training data, however, didn't apply the bottom-of-atmosphere correction.

With frozen model weights, we append and train two additional linear layers for prediction on the Eurosat dataset classes. Even without retraining the extraction feature space we provide, it produces comparable results, see table 6.1. The results directly support the first hypothesis, $H1$, as generalised feature maps produce sufficient results on unseen data, supporting their adaptability to unseen data.

6.5.6 Graph Encoding Comparison

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.

The first experiment computes the measures between each segment, A, and its nearest neighbour, B, in the feature space X (table 6.2) and average that over all the segments. This tests whether the GNN places similar segments together in the feature space.

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. It uses Hungarian matching to create nine pairs between these two sets of neighbourhoods, and utilises

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 6.2: Comparing the similarity between each segment and its nearest neighbour in feature space X.

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

Table 6.3: Comparing the similarity between each segment and its local geographical neighbourhood.

the measures to see how similar they are. The aim of this experiment is to check if a segment in the original image has a similar type of neighbourhood, implicit due to the two graph layers. The feature space should also place a segment close to segments that have similar neighbourhoods as well. For example, a small body of water surrounded by urban segments should be placed near other small bodies of water with similar urban surroundings. Overall, the aim of this test is to evaluate whether 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 the GCN type was better than GAT, see table 6.3. 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. Overall, these models take into account the neighbourhood and texture more than the individual segments or spectral differences.

6.6 Discussion

The embedding spaces generated from the superpixel representation show tighter cluster formation and clearer semantic boundaries than the scene-level embeddings explored in Chapter 5. The earlier work showed unwanted similarity encoding driven by leading textural changes. The segmentation-based representation reduced these effects. The segmentation-level embeddings support fine-grained labelling through the branching methodology. This is visible in Figure 7.4, where a semantic manifold emerges between land cover classes such as urban areas, vegetation, and golf courses. The segmentation and extraction of water sources also show a clear separation between land and water segments. These observations indicate that $H1$ is met. However, the manifold contains outliers. Segments from water with heavy sedimentation sometimes appear similar to land. Smaller water formations, such as lakes, do not always form clear distinctions from neighbouring pixels. The expressiveness of the embedding depends on the number of fuzzy C-means clusters and the quality of superpixel boundaries. C-means with eight classes produced the most stable results in this work, but there is no method to identify the optimal number of clusters for new datasets without running the full pipeline and performing qualitative checks. The SLIC boundaries also vary across features and sometimes overlap homogeneous areas, especially in narrow features such as roads. These conditions limit the consistency of the manifold and show that $H1$ is supported but not unconditional. The introduction of the GNN allowed segments to learn both internal content and relationships with adjacent segments. The visual segmentation analysis and Graph CNN tests show that segments become contextually aware of their neighbours. The neighbourhood and single-segmentation similarity tests show that features become more similar when context is included, compared with similarity measures without it. This is also visible in the embedding space. In Figure 7.4, golf courses cluster as coherent structures despite containing trees, greens, and sand bunkers. Without neighbourhood encoding, these components would likely fragment across the manifold. The same effect appears in the water examples in Figure 6.13. Water segments at land–water boundaries separate from water fully surrounded by water, which is useful for downstream labelling tasks. These results support $H2$. Yet neighbourhood modelling does not improve all features. Small or thin sub-segments can lose distinctiveness when dominated by the surrounding context. This reduces the expressiveness of the embedding in some cases and shows that the effect of neighbourhood modelling depends on the segmentation scale. The examples discussed in this chapter come from multiple chips, this allows an assessment of $H4$. The embeddings group similar segments from different chips and create a shared space

for labelling. This indicates partial support for $H4$. However, chips with large homogeneous backgrounds weaken this behaviour. In some cases, segments from smaller or less prominent features have sub-optimal cross-chip embeddings. Large dominant features can overshadow smaller ones and shift their position in the manifold. This creates fractures in cross-chip consistency and limits the general behaviour expected under $H4$. These limitations do not negate the fulfilment of $H3$. The segments remain semantically meaningful within the manifold even when cross-chip relationships vary. The clusters still reflect interpretable groupings of land cover, water, vegetation, and built structures. This suggests that semantic coherence is maintained at the segment level, even when the chip-level variation reduces global consistency. Overall, the results support the hypotheses but highlight several dependencies. The structure of the embedding depends on superpixel quality, cluster selection, neighbourhood scale, and chip composition. These factors constrain the robustness of the method.

6.7 Limitations and Future Work

The segmentation-based pipeline improves labelling at a granular scale and at the scene level with respect to the structured embeddings. However, several limitations remain. These either arise from the design choices of the pipeline itself or the constraints of graph-based learning.

The pipeline depends on SLIC superpixels to define nodes in the graph. SLIC is efficient but sensitive to spectral homogeneity and compactness parameters. Small linear objects such as roads and hedgerows are split or merged inconsistently. This inconsistency is then propagated through the pipeline, negatively influencing the message passing within the GNN as multiple contextual features are present. Likewise, the labelling at the segmentation level is directly dependent on SLIC being able to extract homogeneous features. If multiple features are present in one image the user has no ability to correct the segmentation boundary or label at the pixel level to circumnavigate the problem.

This limitation is consistent with recent findings showing that classical superpixels exhibit boundary leakage for thin or irregular structures [324, 325]. The adherence of SLIC to a compactness parameter reduces it ability to structure segments that can contort to small narrow feature boundaries, however there are many recent advancements in the field of segmentation that address limitations, see section 6.1.1. Approaches could be made to refine the superpixels via graph-based methods after initial segmentation, to both increase adherence to object boundaries and reduce computational complexity.

Fuzzy C-means provides a weakly supervised target distribution, however, it does not map directly to physical land-cover classes. The centroids are often impeded by illumination, and sensor noise, which can introduce noise into the CNN loss function. This may explain the reduction in latent space quality reported from the graph tests for higher cluster numbers. Future work could address this by locating a more coherent cluster centroid or finding an algorithmic approach to estimating the initial k cluster selection. Alternatively, the model training scheme could be replaced to utilise self-supervised learning methods(SSL), removing the need for pseudo-labels entirely [326]. As explored in section 6.1.2, a self-supervised model may not be as lightweight as our implementation, but could provide a valuable trade-off for efficiency in constructing robust embedding spaces.

The three-layer GNN approach for this work, especially when aggregating 8 neighbours, resulted in over-smoothing. The over-smoothing homogenised embeddings across nodes. Whilst message passing was required to encode spatial information, it needs to be balanced with individual segment representation. This limitation is visible in results where small or spectrally distinct segments (e.g. ponds,isolated roads) lose expressive capabilities. Potential avenues for future research could look to incorporate models such as "GCNII", which looks to mitigate the over-smoothing whilst limiting the increase in computational cost [326], for supervised contexts. More appropriately for unsupervised context is graph learning with SSL addressing over-smoothing by Keriven [327].

Although the embedding space clusters similar features across chips, there are cases where large homogenous backgrounds create fragmentation by clustering independently with similar features in other chips. Chips with large water bodies produce embeddings that are disjoint from structurally small but semantically meaningful smaller water bodies. This limitation is directly related to both the aforementioned message passing over-smoothing and the lack of representation of feature scale. Future work could look to incorporate scale-aware embeddings. A potential solution could assume that one k-hop in the graph, analogous to one message passing layer, could be used for smaller features. This preserves the smaller features by only encoding the most immediate neighbouring context, rather than neighbour or neighbour inclusion seen in 2-hops. Assessing which segments require different scales could be accomplished by similarity metrics (i.e SAM). If the neighbour is spectrally similar, it is likely to require large-scale representation, if not smaller-scale encoding.

Computational scalability is the major limitation of this work; Hungarian matching is the most computationally expensive component, $O(N^3)$. This complexity is compounded when

considering a complete similarity matrix of $N = 42^2$ chips per tile. Future work could look to approximate the similarity between chips then compute matching to create a sparse similarity matrix. However, with a sparse matrix, further evaluation would have to be taken regarding how many k similar chips are utilised. Alternatively, an unsupervised deep graph matching algorithm could be utilised; recent works by Tourani et al. show how to match graphs based on key points in images using *cycle consistency* [328]. The proposed method can be applied to any GNN backbone replacing classic matching solvers.

The evaluation of the section has limitations, the works are primarily focused on qualitative analysis. The works are limited by resources and accessibility to expert labellers, making large-scale validation difficult. Whilst we assess the manifold via unsupervised metrics and nearest neighbour testing, in addition to testing segmentation with respect to their neighbourhood, the methodology could be expanded to compare latent spaces to embeddings. The metrics(SAM, SSIM, et.c.) could be considered as pseudo-similarity labels to compute the Neighbourhood Hit Rate(NHR) or mean average precision of the nearest k neighbours(mAP@k) [162, 329]. NHR measures the nearest neighbours that match the true similarity signal and mAP@k ranks the quality of k neighbours in high-dimensional space.

6.8 Conclusion

In this chapter we present a novel convolutional and feature extraction method for image comparison at the scene level and at a segmentation level for Sentinel 2 remote sensing data. We trained models in an unsupervised fashion by utilising unsupervised fuzzy clustering to produce a supervised pipeline. This pipeline can be trained on a small quantity of training data, a single tile, and generalised to multiple geographical locations. We utilised a U-Net for texture and graph neural networks to encode neighbourhood information. By clustering the resultant encodings within a labelling tool, we demonstrate the effectiveness of our pipeline in comparison to previous works in chapter 5. Namely, improving clustering and removing rotational dependencies. In addition, we show that complex features can be labelled at the segmentation level within the application.

Chapter 7

Interdisciplinary Unsupervised Labelling: Abundances and Feature Encoding

In Chapter 6, we demonstrated that superpixel representations, combined with CNN features and GNN message passing, can generate meaningful segment embeddings for interactive labelling. However, the efficiency of that pipeline depends on the quality of the pseudo-labels utilised during training for both the CNN and the GNN. In Chapter 6, pseudo-labels were computed using fuzzy C-means clustering, a standard method for clustering pixels with soft assignments. Whilst effective, C-means clustering provides no physical interpretation of the resulting clusters and does not exploit domain-specific properties of spectral remote-sensing images. This limitation motivates the exploration of alternative unsupervised pseudo-label pixel assignments.

In this chapter, we therefore replace fuzzy C-means with the N-FINDR [75] algorithm. The N-FINDR algorithm is a well-established endmember extraction algorithm used for spectral unmixing, see Section 2.6. In comparison to C-Means, which defines an arbitrary cluster centroid, N-FINDR identifies the most spectrally "pure" pixels in the dataset and models each pixel as a mixture of these endmembers. The mixture of endmembers assigned to each pixel is referred to as abundance vectors. Abundance vectors are analogous to the membership vectors produced by C-means, however, they correspond to meaningful physical materials.

N-FINDR and related algorithms are routinely used as alternatives to clustering for segmentation, representation learning, and pixel grouping cite REN2026100035. In particular,

their use is well observed as they preserve the spectral data and capture material composition, rather than arbitrary assignment. We propose replacing our C-means within the pipeline of Chapter 6 with N-FINDR, as it is theoretically valid and supported in the literature.

7.1 Literature Review

Classically, in remote sensing, the intensity of each returned wavelength at a sensor can be modelled as a function of the sensing, atmospheric, and material properties. If the material properties can be identified, then a library of known reflectance values can be used to classify the unseen data. Canonically, these extracted material wavelengths are referred to as endmembers. We have touched upon these concepts in 2.6. Due to the low spatial resolutions of satellite imagery, there can be multiple endmembers within a single pixel. Alternatively, the material can be a homogeneous combination of multiple endmembers regardless of image spatial resolution; for example, sand and water at a beach.

In the case of each pixel containing a non-homogeneous material, we must determine the proportion of each endmember present, known as an abundance, which can be achieved through linear spectral unmixing. Most geometry-based endmember algorithms compute on spectral space and assume the linear mixing model(LMM) [330]. Under the linear mixing model (LMM), a pixel's spectrum is a convex combination of endmember spectra. The abundance values must be non-negative. In the fully constrained case, they must also sum to one. These constraints ensure the mixture proportions are physically meaningful and represent the full composition of the pixel. For homogenous mixtures, we must infer there is a non-linear combination [72]. The general input of unmixing algorithms is the observed pixel spectrum. They use the input to calculate endmember sets and output abundance vectors for every pixel.

In general, there are two approaches to finding endmembers: "internal maximum volume models" and "external minimum volume models" [331]. The former consists of algorithms such as Pixel Purity Index(PPI) [73], N-FINDR [75], simplex growing algorithm (SGA) [332], Vertex Component Analysis (VCA) [333], Sequential Maximum Angle Convex Cone(SMACC) [334] and Alternating Volume maximisation(AVMAX) [335]. The latter consists of Minimum-Volume Simplex Analysis (MVSA) [336], Minimum-Volume Enclosing Simplex (MVES) [337] and Minimum-Volume-Constrained Non-negative Matrix Factorisation (MVC-NMF) [338]. Sun et al. classify these algorithms by whether they rely on the pure-pixel assumption; some pure pixels exist in the given dataset cite rs71215834. This assumption can

be unfavourable for images with noise and mixed materials. Another key differentiation is that most algorithms require some pre-selection of parameters, with the exclusion of N-FINDR.

Most of these algorithms are looking for points lying on the outer extreme of the data distribution. "Outer extreme" does not refer to spatial location in the image, but to the position of a pixel's reflectance vector within the high-dimensional spectral space. This is the key difference between centroid-based algorithms, such as C-means utilised in Chapter 6, which assume a central position rather than boundaries. Within the LMM, pure pixels correspond to the vertices of a simplex, and mixed pixels are a combination of those vertices. Because of the abundance constraints, all valid mixtures must lie inside the convex hull formed by the endmembers. The endmembers themselves lie at the extreme points of this hull.

Three key reasons define the assumption of pure pixels on the extreme points:

- Spectrally distinct pixels form the edge of a distribution in spectral feature space with mixed pixels between the pure pixels [330].
- Mixtures typically follow a convex/simplex geometry in spectral space [330, 339].
- Mixed pixels are a proportion or sum of endmembers, and they must, therefore, lie within the convex hull formed by the endmembers [339, 340].

7.2 Endmember Extraction Algorithms

Endmember extraction algorithms share a common objective: given a multispectral or hyperspectral image represented as a set

$$P = \{p_1, p_2, \ldots, p_N\}, \quad p_i \in \mathbb{R}^B,$$

where B is the number of spectral bands, the goal is to estimate a set of K endmembers

$$E = \{e_1, e_2, \ldots, e_K\},$$

which correspond to extreme spectral signatures forming the vertices of a simplex that encloses the data.

We use the following notation:

- P: set of all pixel spectra,
- p_i: the i-th pixel spectrum,

- B: number of spectral bands,
- K: number of endmembers,
- $E = [e_1, \ldots, e_K]$: endmember matrix,
- $V(E)$: simplex volume.

All methods aim to estimate endmember candidates lying on the convex hull of the dataset.

7.2.1 Pixel Purity Index

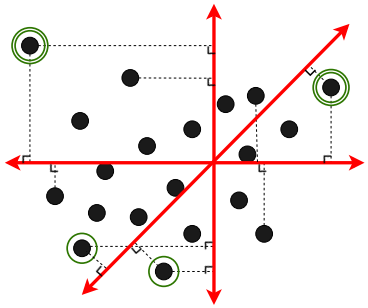


Figure 7.1: Example of Pure Pixel Index skewers. The red lines are the skewers. The points on the extrema are circled in green. For pixels with multiple "votes" have additional green circles.

PPI generates many random vectors, known as skewers, in a B-dimensional space, where B represents the number of wavelengths within the data. Each data point is projected onto the skewer, with the two extrema points recorded [73]. After each skewer is processed, the extrema points with the highest tallies are considered purest, and the number of times a particular point has been recorded is its PPI.

7.2.2 N-FINDR

N-FINDR tries to maximise the volume of a simplex in the feature space of the pixels P [75]. The points utilised to signify the vertices of the simplex are considered the purest and are taken to be the endmembers. As the simplex is maximised, the points will lie on the outer edges of the dataset. The algorithm is defined as:

1. Reduce the original multi or hyperspectral image to n dimensions utilising Maximum Noise Fraction (MNF), where $n+1$ is the number of endmembers to be extracted.

2. Initialise a set of vectors $(e_0^{(0)}, e_1^{(0)}, e_2^{(0)}, ..., e_n^{(0)})$.

3. Calculate the volume V of the simplex produced by the vectors:

$$V(E^{(0)}) = \frac{|\det(E^{(0)})|}{n!}.$$

where

$$E^{(0)} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ e_0^{(0)} & e_1^{(0)} & e_2^{(0)} & \dots & e_n^{(0)} \end{bmatrix}.$$

4. For every pixel $p_i \in P$, replace each endmember candidate j with the pixel as a vertex. If the computed new volume V is larger, then replace $e_j^{(i-1)}$ with p_i.

5. The algorithm concludes when all pixels have been tested for each vertex position.

7.2.3 Simplex Growing Algorithm

SGA is derived from and inspired by N-FINDR, where a simplex is used to extract endmembers [332]. The key difference is that SGA expands the convex hull iteratively whereas N-FINDR grows the simplex. The algorithm begins with two vertices, and with each iteration, it grows the simplex by adding a vertex, terminating once all endmembers are identified.

1. Initialisation of the first vertices is computed by calculating PCA or MNF to two dimensions. With a random target pixel t selected, the first endmember e_1 is found using

$$e_1 = \max_r \left| \det \begin{bmatrix} 1 & 1 \\ t & r \end{bmatrix} \right|.$$

2. We now calculate the volume V, iterating over each sample point r:

$$V(e_0, e_1, e_2, \dots, e_n, r) = \frac{|\det([e_0, e_1, e_2, \dots, e_n, r])|}{n!}.$$

3. Iteratively find the remaining endmembers e_{n+1} that maximise V until the number of endmembers is reached.

7.2.4 Vertex Component Analysis

VCA, like most other algorithms, looks to create a convex hull [333]. There are two main constraints for understanding this algorithm. Firstly, all abundances are a fraction, and the

fractional abundances must sum to one for any given pixel. The algorithm selects orthonormal vectors for each sequential iteration p as the number of endmembers to be extracted. The maximum value on each projected orthonormal vector constitutes a new endmember.

- **Input:** p (number of endmembers) and $z \equiv [r_1, r_2, \ldots, r_n]$.
- **Initialise:**
  - $M := 0$ { $L \times p$ estimated mixing matrix }
  - $f := [1, 0, \ldots, 0]^T$ { Initial projection vector }
- **For** $i = 1$ to p **do**
  1. Compute the projection:
  $$y := f^T z$$
  2. Find the index of the maximum absolute value:
  $$k := \arg \max_{j=1,\ldots,n} |y(j)|.$$
  3. Assign the selected endmember:
  $$M_{:,i} := z_{:,k}.$$
  4. Generate a new vector f orthonormal to the span of previously selected endmembers:
  $$f := \text{orthonormal}(M_{:,1:i}).$$
- **End for**

7.2.5 Alternating Volume Maximisation

Alternating Volume Maximisation is another methodology derived from N-FINDR. The commonality is finding a convex hull that contains the datapoints, with endmembers forming the vertices. The method of calculating the maximum via the determinant is the same. The difference is that, as the name suggests, each iteration finds an optimal set of endmembers while keeping the rest the same. Whilst N-FINDR is a greedy approach, this method follows a structured optimisation, keeping computational costs low with less sensitivity to initialisation. The algorithm is detailed in [332].

7.2.6 Convex Cone Analysis

CCA assumes that pure pixels and endmember candidates lie on the outer edges of the sample data and relies on non-negative values produced by physical reflectance spectra [341]. Discrete reflectance spectra can therefore lie within a convex region formed by non-negative components. The steps taken by CCA are:

1. Given a matrix S composed of samples in columns and wavelengths in rows, derive the correlation matrix
$$C = S^T S$$
. Apply singular value decomposition: $C = PDP^T$, where P contains eigenvectors.

2. Derive a set of linear equations:
$$x = [p_1 \ \dots \ p_c] \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_{c-1} \end{bmatrix}.$$
Corners are found by minimising x subject to $x \geq 0$.

3. If $c = 1$ then the solution is p_1. Otherwise, search all combinations of $a_1, \dots, a_{c-1}$ satisfying the condition $x = 0$.

7.2.7 Minimum-Volume Simplex Analysis (MVSA)

MVSA assumes that the data contain a pure pixel for each endmember [336]. The algorithm solves a hard non-convex optimisation problem where endmembers are adjusted toward a local minimum while containing all datapoints. Initialisation (often via CCA or VCA) is crucial to achieving a good solution.

7.2.8 Minimum-Volume Enclosing Simplex (MVES)

MVES aims to find the smallest simplex that encloses the source points [337]. The determinant is again used to calculate simplex volume, but the optimisation is solved through linear programming. Minimum-Volume-Constrained Non-Negative Enclosing Simplex is a variation with the constraint that all endmembers lie in the non-negative orthant.

7.2.9 Comparison

Algorithm	Optimization Type	Approach	Public Python Implementation	Computational Complexity
PPI	Heuristic / Search	Selects pure pixels based on projections onto random skewers; extrema accumulate purity votes.	Yes (via `spectral` package)	**O(S · n)**: S skewers, linear scan per skewer
N-FINDR	Geometric Optimisation	Iteratively replaces simplex vertices to maximise simplex volume using determinant updates.	Yes (via `spectral` package)	**O(n · K^4)**: n pixels, K vertices, volume via $K \times K$ determinants
Simplex Growing Algorithm	Geometric / Greedy Optimisation	Grows a simplex outward by adding the pixel that maximises the increase in simplex volume.	No (common variants only)	**O(n · K^3)**: determinant evaluation per candidate expansion
VCA	Projection-Based (Geometric)	Uses random projections and orthogonal subspace updates to identify extreme-point vertices.	Yes (various implementations; MATLAB original)	**O(nB^2 + K n)**: whitening + projection search
AVMAX	Alternating Optimisation (Block-wise)	Alternates updates for one endmember at a time while fixing others; maximises simplex volume.	No (limited public implementations)	**O(n K B^2)**: repeated matrix operations in B-dimensional space
Convex Cone Analysis	Geometric Optimisation	Identifies cone vertices under non-negativity constraints using the eigenstructure of the correlation matrix.	No	**O(B^3)**: dominated by eigen-decomposition
MVSA	Convex Optimisation	Minimises the volume of a simplex enclosing the data; requires iterative constrained solving.	No	**O(K B^3)**: optimisation in B-dimensional space
MVES	Convex Optimisation (LP-Based)	Finds minimum-volume enclosing simplex via linear programming formulation.	No	**O(K B^3)**: similar to MVSA due to matrix operations

Table 7.1: Comparison of Endmember Extraction Algorithms with Computational Complexity

A comparison has been made of endmember extraction algorithms by Plaza et al. [342]. They compare simulated data collected via endmember libraries from the Airborne Visible and Infrared Imaging Spectrometer (AVIRIS) with simulated data at increasing steps of signal-to-noise ratio, with the additional constraint of no non-linear mixtures. The different initialisation conditions also compound the complexity of comparing algorithms. They found that methods combining spatial and spectral information, such as Convex Cone Analysis (CCA) or Automated Morphological Endmember Extraction (AMEE), were more accurate [341].

As with most algorithms, there are many models with parameter optimisation, such as autoencoders [343, 344] or genetic algorithms such as artificial bee colonies [331]. The classification and subsequent abundance values of a given dataset or site must be produced to train the models. The loss function then takes into account the original data. To compare the abundances, standard evaluation metrics include root mean squared error (RMSE) and spectral angular distance (SAD).

The comparison in Table 7.1 highlights key insights into the approach and limitations of existing endmember extraction approaches. Internal maximum-volume algorithms such as PPI, N-FINDR, SGA and VCA depend on the "pure" pixel assumption [330, 331], which presumes that at least one pixel corresponds to each endmember. This assumption works well for hyperspectral data but breaks down for multispectral imagery like Sentinel-2, where mixed pixels

dominate. Several algorithms also show sensitivity to noise (e.g., PPI [73]) and to initialisation (e.g., N-FINDR and MVSA [336]), and some require user-defined parameters such as the number of skewers or projection vectors. External minimum-volume models such as MVSA, MVES and MVC-NMF relax the pure-pixel assumption but involve solving more complex optimisation problems and may converge to sub-optimal local minima without careful initialisation [336, 337].

7.3 Motivations

Motivations section to illustrate the key choice of replacement in the previous pipeline. These observations identified in the previous section motivate the methodology adopted in this chapter. N-FINDR provides a good balance of simplicity and interpretability. Additional the algorithm needs minimal tuning, identifies reliable pure pixels, and produces abundance vectors that reflect real material composition [75, 345]. Replacing the C-means pseudo-labels from Chapter 6 with N-FINDR abundances overcomes a key limitation of C-means. Where C-means produces arbitrary clusters with no physical meaning, N-FINDR's abundance vectors are physically grounded. This combination of classical unmixing with modern neural models is the main novelty of our approach.

This is our hypothesis given the motivation:

H1: Physically meaningful abundance vectors derived from N-FINDR will provide a more stable latent space than C-means cluster assignments, leading to improved segmentation embeddings.

7.4 Methodology

This chapter utilises the same pipeline considered in chapter, 6, for learning and extracting features from remote sensing imagery. We now combine the computer vision pipeline from Chapter 5 with a dedicated remote sensing algorithm and approach, substituting a part of the pipeline with a classical remote sensing algorithm. We test whether the interdisciplinary approach can offer any new perspectives or increased model performance compared to a pure CV approach. Therefore, we substitute modules in the pipeline as follows. We substitute C-means for an endmember extractor, N-FINDR, for it is a domain-based algorithm. The N-FINDRs methodology is easier to implement compared to other algorithms we have considered, as most are extensions of N-FINDR. Its implementation is also attributed to the lower computational

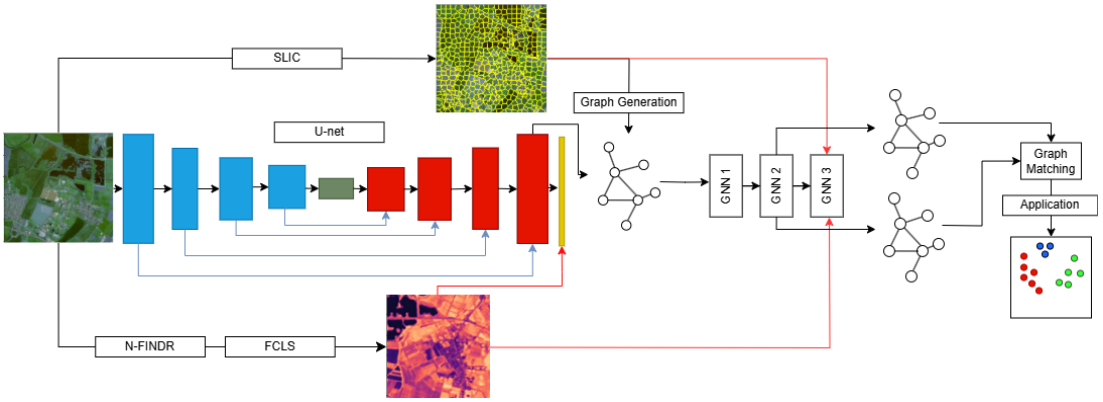


Figure 7.2: Pipeline for our methodology, blue boxes denote the constricting layers of the U-Net, with red denoting the expansive layers. Yellow is the predictive layer. Blue lines indicate a skip connection. Red lines indicate the use of data within the loss function. Similar to Chapter 7, however, C-Means has been replaced by NFINDR and FCLS.

complexity compared to other algorithms. It is also one of the few fully automated algorithms that require only the number of endmembers to extract as a parameter. For all of these reasons, it has seen popularity within the RS field. In conjunction with endmember extraction provided by N-FINDR, we use a linear unmixing model to assign abundance value to each pixel based on the calculated endmembers. C-means would assign a fractional portion to each pixel that would sum to one.

To compare the centroid to endmember substitution, we utilise a fully constrained linear unmixing model, which sums each fractional abundance to one [346]. As N-FINDR utilises the minimum noise fraction transformation, we have an additional constraint: only 12 endmembers can be extracted. Therefore, we also compare the results of 12 c-means within the results. Given any tile, there are more than 12 materials present, especially when including the spectral signatures of non-homogeneous materials. Therefore, we still retain an approach that pseudo-labels the pixels for generalisability. We had to adjust the CNN loss function by assigning more importance to the dice loss than to the binary cross-entropy, with respective weightings of 2 and 0.5, to improve training stability. For intuition, if we consider endmembers as cluster centroids, then we know that these new centroids will lie near the outer distribution as defined by a convex hull. Points which are near the centre of the sample space will consist of minute fractional differences when assigning membership. Binary cross-entropy on minute differences between the targeted pseudo-labels leads to large fluctuations and instability.

Comparisons are made on both the convolutional neural networks(CNNs) and the feature space,X, extracted from the second layer of the graph neural networks(GCNs). The CNNs

are tested against the EuroSat dataset to validate the activation maps independent of the GCNs. Testing the feature spaces produced by each graph, X, is conducted through the visual labelling application and the graph tests for similarity. The graph tests are conducted before any dimensionality reduction is performed, thereby ignoring the penalty of embedding high-dimensional features into two dimensions. At the same time, visual inspection is done with the limitation of a two-dimensional embedding, which is highly dependent on the structure of feature space,X, both in similarity and shape. The CNN tests assess the activation maps extracted by testing their utility in classifying another benchmark dataset. The graph tests provide insight into the mapping of image segmentations into the high-dimensional feature space provided by each GCN. The visual application is finally an insight into how informative the two-dimensional embeddings are, considering labelling as a primary objective.

7.5 Results

Here we compare the differences between the pipelines from the previous chapter, cmeans, and this pipeline.

7.5.1 N-FINDR vs C-Means pipelines for Remote Sensing Labelling Application

We kept all interactions in the labelling tool the same as in chapter 6. The user can load a specific dataset with the initial projection of points, each representing a 256 by 256 pixel image we have referred to as chips. This initial projection enables the user to select chips with features of interest without having to filter through the numerous segmentations that comprise each chip, which is approximately 500. Exploration is conducted within a region that can be highlighted, resized, and translated over points to display its contents in a coordinated window; see examples in Figure 7.3. Utilising the highlighted region, the user can also assign labels to each point. If the user wishes to select smaller or sub-features from labelled chips, they can branch them. Branching creates a self-contained environment of selected images, which can be projected into two dimensions individually, with the added functionality of merging labels back to the main branch. Within branched environments, the user can also project each segmentation subset, with exploration and labelling now occurring at the segmentation level; see examples Figures 7.4 and 7.5.

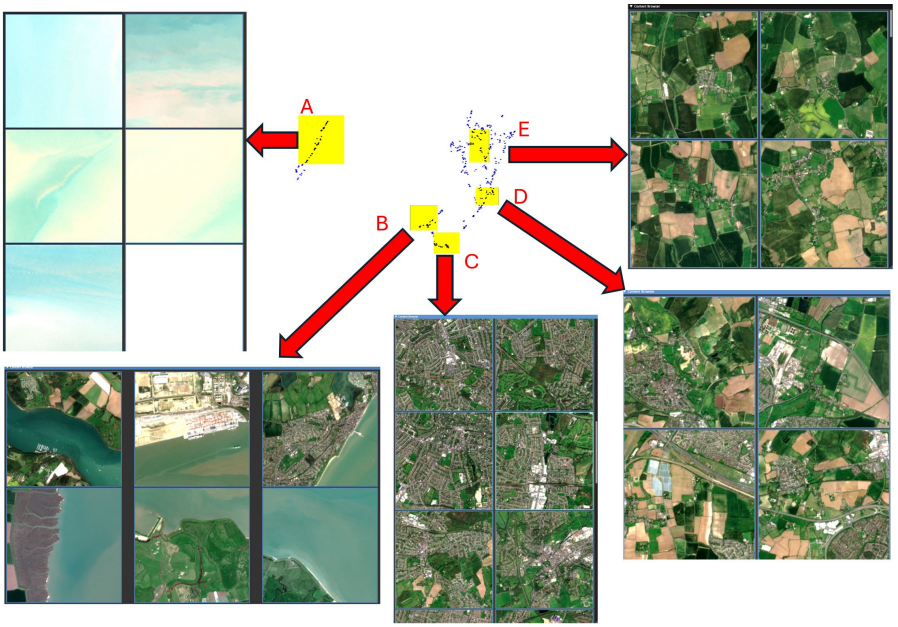


Figure 7.3: Example of exploring the chip level embedding space, produced by N-FINDR with 12 endmembers, by interactively highlighting different regions. A is a cluster predominantly with water. B is a mixture of coastal, or large water features with land. C is dense urban environments. With D and E showing the gradual introduction of more vegetation.

Firstly, we explore the manifolds computed from the 12 endmembers; see Figure 7.3. The construction of the space, predictably, emphasises the separation of land and water features, the most distinct features. With A containing the majority of water and C,D and E consisting of land-based features. Sample B is a combination of the two, with its placement more central. Within land features, we can see a distinct evolution of highly man-made urban features to more vegetative samples, evolution from C to E. Comparatively to the solution in chapter 6 Figure 6.10, we see no significant changes to the construction of the embedding space with respect to the evolution of the manifolds, given the stochastic nature of UMAP.

Previously, we explored four images, Figure 6.12, which show the separation between urban and vegetation areas, as well as the selection of points indicating a golf course. We have reproduced the results with our new pipeline, Figure 7.4. Notably, all features can be selected with ease, where urban segments lie horizontally above and more vegetative samples lie below. Except for the top-left samples, which also contain vegetation. The projected space from X, reduced to two dimensions, is more convoluted for the separation of the examples shown compared to C-Means. This is more so exacerbated by the use of a rectangular selection tool where the grouping does not conform to singular selection. In total, we see the same

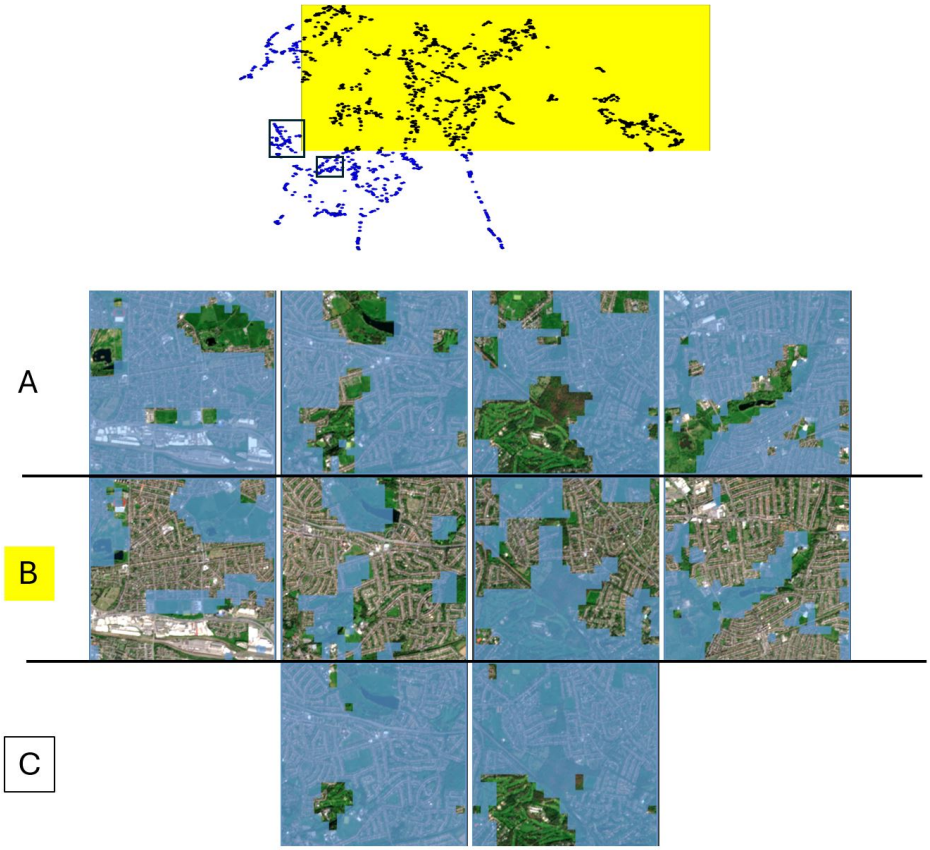


Figure 7.4: Example of labelling segments of four different images. A showing vegetative samples, the unselected points. B, highlighted, show the urban segments. C is an example of golf courses, shown through a bounding box.

functionality as before; however, the extraction of features is more disjointed. For example, the extraction of golf courses is divided into two separate selections.

Lastly we show an example of water selection from four images containing a river, coastal water and lakes, see Figure 7.5. The highlight of this selection has been purposely placed on apparent clusters towards the edge of the projection. This interaction method is identical to that shown in 6.13, which was presented for comparison. The clusters contain segments that are primarily composed of water, except for some outliers and the right-most example, which includes lakes. The lakes example, as before, seems to be the most difficult to cluster. Additionally, when examining the coastal scene, the segments are confined to a line along the coast. This may be a causal factor introduced by the GNNs clustering similar neighbourhoods of a particular segment, caused by message passing.

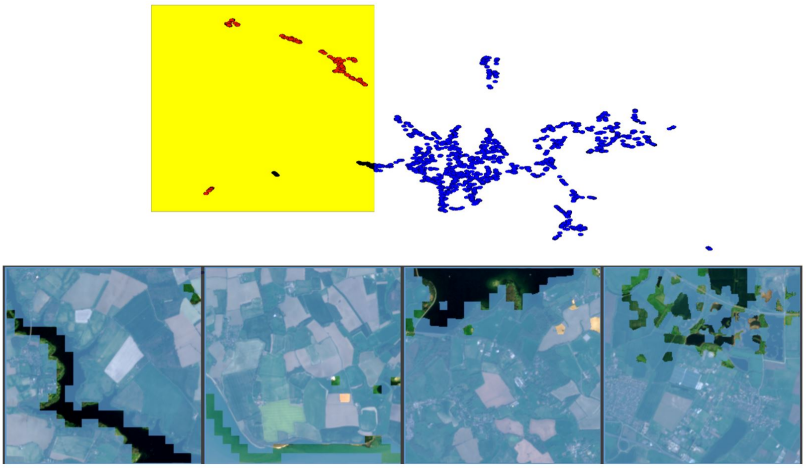


Figure 7.5: Example of finding similar water segments. Variation of rivers, lakes and coastal waters.

In this section, we have visually identified and compared clusters in the two-dimensional embedding space provided by UMAP. The results for chip comparison are similar to C-means, if not better, with some more distinct classes. The segmentation features, although informative, considering they are utilised to construct chip similarities, have a more complex structure in high dimensionality. This complexity is only in reference to the dimensionality reduction of the high-dimensional structure for producing an easy labelling experience.

7.5.2 CNN Tests

Within this section, we test the CNN as an independent component of the pipeline. This test is identical and reproduced for the new CNNs from section 6.5.5. To reiterate, we test the data on the EuroSat dataset, which consists of 10 different classes from locations around the European Union [2]. EuroSat data is compiled from the top-of-atmosphere preprocessed data, whereas our training data from SWED, is bottom-of-atmosphere, adds to the complexity of the task. The model weights are frozen with the addition of 2 learnable linear layers for classification. This approach to testing models by freezing weights and training only a lightweight classifier on another dataset can be seen in literature [266, 326]. The results reported in Table 7.2 show the benchmark from the Eurostat dataset and our CNN models; all models were trained on an 80-20 train-test split. The Eurosat benchmark has been trained on the EuroSat dataset.

C-means approached an apparent maxima near 8 classes when trained on a tile from SWED and tested on EuroSat. In contrast, when utilising 12 endmembers automatically selected by N-

Method	Accuracy
EuroSat ResNet-50	96.43
C=2 linear layers	93.67
C=8 linear layers	95.16
C=12 linear layers	93.24
C=18 linear layers	92.87
A=2 linear layers	91.06
A=8 linear layers	94.83
A=12 linear layers	**97.47**

Table 7.2: Respective accuracies for each model tested, where C is the initial C means clustering number and A is the number of endmembers. Results labelled EuroSat were taken from [2].

FINDR, we report a higher accuracy than the original benchmark from the EuroSat dataset. We can therefore infer that for the samples present in Eurosat, the activation maps produced by N-FINDR are more informative. Given that our model is pre-trained with no fine-tuning towards the new dataset, from multiple regions, or for top-of-atmosphere reflection, we can conclude that our representations of RS images extracted from the CNN are a good generalisation for downstream tasks.

7.5.3 Graph Tests

As before, section 6.5.6, we have trained two types of models, a GCN and GAT, with similar parameters and hidden features in the final extraction layers. We test each segment's closeness in the feature space provided by each model and labelling scheme. The similarity is determined by textural contrast with GLCM and LBP. SSIM measures the structure, contrast and luminance. Finally, SAM is purely based on the mean spectral differences between each segment. We compare the similarity of embedded segments within each model and then a second test also to incorporate their geographical neighbourhoods.

Table 7.3 reports the results for comparing a singular segment to its closest segment in feature space X produced by the second layer of each graph neural network. For all similarity measures, the GCN trained on 12 classes pseudo-labelled with N-FINDR performed the best. In general, GCNs outperformed their GAT counterparts. Comparing the two pseudo-labelling strategies, we find that fewer C-Means classes outperform similar abundance endmembers.

Similarly, when reporting the similarity between segments and their geographical neighbourhoods, table 7.4, 12 endmembers have the best feature space concerning similarity. This

Model	Labelling	GLCM↓	LBP↑	SSIM↑	SAM↓
GCN 2	C-Means	16.2770	0.8571	0.9120	0.3321
GCN 8	C-Means	15.4685	0.8716	0.9287	0.2887
GCN 18	C-Means	15.5322	0.8671	0.9243	0.3017
GAT 2	C-Means	16.0025	0.8633	0.9201	0.3112
GAT 8	C-Means	15.5676	0.8704	0.9256	0.2942
GAT 18	C-Means	15.7663	0.8673	0.9237	0.2992
GCN 2	N-FINDR	16.8789	0.8446	0.8992	0.3693
GCN 8	N-FINDR	15.3052	0.8749	0.9326	0.2789
GCN 12	N-FINDR	**14.2941**	**0.8931**	**0.9530**	**0.2230**
GAT 2	N-FINDR	16.5570	0.8516	0.9078	0.3463
GAT 8	N-FINDR	15.2665	0.8759	0.9329	0.2752
GAT 12	N-FINDR	15.1873	0.8799	0.9374	0.2619

Table 7.3: Comparing the similarity between each segment and its nearest neighbour in feature space X produced by each GCN.

Model	labelling	GLCM↓	LBP↑	SSIM↑	SAM↓
GCN 2	C-Means	14.4610	0.8983	0.9589	0.2152
GCN 8	C-Means	13.7904	0.9034	0.9624	0.2064
GCN 18	C-Means	13.7278	0.9033	0.9620	0.2066
GAT 2	C-Means	14.2259	0.9000	0.9560	0.2122
GAT 8	C-Means	14.0145	0.9018	0.9613	0.2092
GAT 18	C-Means	14.0557	0.9013	0.9611	0.2100
GCN 2	N-FINDR	14.6950	0.8964	0.9573	0.2185
GCN 8	N-FINDR	13.6778	0.9041	0.9630	0.2049
GCN 12	N-FINDR	**12.7270**	**0.9114**	**0.9674**	**0.1925**
GAT 2	N-FINDR	14.5575	0.8972	0.9580	0.2171
GAT 8	N-FINDR	13.8970	0.9029	0.9621	0.2078
GAT 12	N-FINDR	13.9116	0.9029	0.9623	0.2075

Table 7.4: Comparing the similarity between each segment and its local geographical neighbourhood.

test is identical to the single-segment comparison above. However, we also compute the similarity between the eight closest geographical neighbours and segments. Instead of comparing all nine segments to every other segment, we opted to reduce the matching to a one-to-one computation. Utilising the Hungarian matching algorithm, we can find the most similar segments between each set of 9 segments in feature space X. For example, if we take a coastal segmentation that represents a beach and the closest segment in space X, both segments are close to other segments with sand, water or rocks. The Hungarian matching will pair each segment with

another, so if four neighbouring segments containing water are present in the initial sample, the corresponding similar segment should also have only four neighbouring water segments. If the neighbourhoods are not exact, then the comparison metrics will indicate this dissimilarity. To compare both single- and multi-segment analyses, reported in the tables above, we average the number of comparisons made. We demonstrate that GCNs trained with 12 endmembers yield the optimal feature space for both singular as reported in the tables above, we calculate the average segment and neighbourhood comparison, with models generally accounting for the latter more effectively than the former.

7.6 Discussion

New section to discuss results in comparison to the hypothesis and its limitations. As only one change was made these can be summarised into one section. The aims of this chapter was to test the hypothesis that physically meaningful abundance vectors extracted using N-FINDR would provide a more stable latent space. This latent space was in comparison to works in the previous Chapter 6 where C-means was utilised instead for cluster assignment. Our work shows that N-FINDR has led to improved segmentation embeddings. Across the quantitative experiments, the hypothesis is strongly supported.

The EuroSat experiments in section 6.5.5 and table 7.2 show that models trained with N-FINDR abundances can match or exceed the performance of those trained with C-Means labels. In particular, we found that 12 endmembers (A=12) provided the best results. The work does not extend beyond 12 endmembers due to the constraints of the algorithm. These results were on a different dataset, originating from different sensing conditions and preprocessing steps, highlighting the generalisability of the feature maps. This shows that feature maps from N-FINDR are able to generalise well. However, the Eurosat dataset has images within regions of similar climates to our training set, and therefore, we can not conclude that the features generalise over differing climates and geographical regions.

One limitation of N-FINDR is that it assumes the existence of approximate pure pixels in the scene. This assumption does not hold for multi-spectral scenes where mixed pixels are common and non-linear effects can arise. Therefore, much like C-Means, N-FINDR in our context is simply an analogous representation of pure pixels that better represent the data. As a result, the networks cannot learn true material compositions because they are not modelled in the loss function. Work could be undertaken to extract more suitable endmembers.

The graph-based tests further support the hypothesis. Tables 7.3 and 7.4 compare the similarity of the latent space X at the segmentation level, both for nearest neighbours and the geographical neighbourhood of a segment. For both tests, our pipeline with 12 abundances is shown to consistently achieve better scores across all metrics, which indicates better similarity in high-dimensional latent space both texturally and spectrally between segments. The results are positive for both pairs of segments in the high-dimensional space and their neighbourhoods. However, much like the C-means models, the N-FINDR model also produces better results when neighbourhoods are accounted for; this has been seen to convolute smaller features.

The qualitative analysis using the interactive tool complements the quantitative results. At the chip level, there is little to no qualitative change for a perspective user, aside from some class transitions, such as urban to vegetation, that appear slightly more defined and sharper. At the segmentation level, Figures 7.4 and 7.5 show that the tool still enables consistent discovery of meaning segment clusters. However, these clusters exhibit some fragmentation and require additional effort for users to label them. This reflects some disparity between the positive results in the high-dimensional space and the embeddings a user interacts with. This highlights a limitation of the works for both the manifold learning techniques and the latent space graph tests. The graph tests do not explicitly check for clusters in the high-dimensional space, however, they rely on k nearest neighbours. A cluster in a high-dimensional space could be tested. Also, much like chapter 6, the quality of the manifold learning techniques could be tested. As the work, in Chapter 6, utilises the same pipeline with minor changes many of the limitations concerning the CNN and GNN still apply to this work.

A major limitation of this work is the inability of expert users to interact with the tool and provide quantitative and qualitative feedback due to resource constraints. The works would have been able to answer more hypotheses if user interaction had been explicitly tested, especially annotation speed.

7.6.1 Conclusion

In this chapter, we have implemented a pseudo-supervised pipeline that relies on unsupervised labelling provided by endmember extraction and their abundances. The methodology is inspired by the assumption within the remote sensing field that all pixels within an image are either an endmember or consist of fractional proportions of endmembers. To maintain a fully unsupervised methodology, we extract the endmembers from the dataset by assuming the presence of pre-existing pure pixels. Our pipeline consists of three main algorithmic blocks: abun-

dance calculation (via linear unmixing), textural extraction (via CNNs) and neighbourhood encoding (via GCNs). Endmember extraction is conducted by N-FINDR, and fully constrained least squares calculates abundances. The textural features are from a Unet architecture utilising the final hidden activation maps. Lastly, the graph networks allow some neighbourhood segment encoding. The simple linear iterative clustering algorithm is employed for segmenting the initial image. We compare this methodology with our previous pipeline.

We have demonstrated that utilising traditional endmember extraction algorithms for pseudo-labelling is more beneficial for remote sensing data than purely clustering-based approaches seen in computer vision, within our pipeline. The convolutional neural network's image representation tested on the EuroSAT dataset achieved significant improvement, surpassing the benchmark set by the dataset's authors. As the CNN is only pre-trained and not fine-tuned, the representations can be said to provide discernible information. The feature space of each segment, as produced by the graphs shown, is also improved when considering the tested metrics. Each segment is closer to similar segments and their respective geographical neighbourhoods. The high-dimensional encodings are better than before; however, projecting these spaces to two dimensions yields a slightly more complex projection in terms of labelling ease. For chip-based comparisons, the two-dimensional projection proved informative and comparable to C-means, if not superior. However, when considering each segment in the labelling tool, whilst still effective, it was slightly harder to find features of interest due to their fragmentation. This, for future work, clearly indicates that we need an implicit or explicit way to affect where each feature is encoded. For many tasks, this is typically done using methods such as triplet loss. In addition, the segmentation algorithm used could be replaced or optimised to enforce more homogeneous segments, as most currently contain multiple features.

Chapter 8

Conclusions and Future Work

Contents

8.1 Overview

This thesis addressed the challenge of unsupervised and weakly supervised labelling in remote sensing, focusing on three major problems identified in Chapter 3:

- 1: the cost and scarcity of labelled datasets, especially for domain-specific challenges.
- 2: the difficulty of navigating large-scale unlabelled satellite archives.
- 3: the lack of tools that support interactive dataset construction.

There are several challenges that must be overcome to fully utilise remote sensing imagery in industrial applications. Firstly, a large volume of imagery data is generated every day, and only a limited number of Remote Sensing experts are available to conduct thorough analysis. Secondly, the combination of the complexity of the Earth's surface and atmosphere and the particularities of the remote sensing technologies creates RS imagery that is complex and difficult to interpret. Machine Learning techniques enable scalable analysis of imagery and, consequently, the extraction of actionable information. However, a significant bottleneck in using machine learning for remote sensing image analysis is the paucity of diverse, large, and

accurately labelled datasets. Therefore, the main objective of this thesis has been to investigate how data visualisation and unsupervised machine learning techniques can be combined to enable users to create large-scale, labelled remote-sensing datasets for subsequent analyses.

We combined methodologies from unsupervised machine learning, data visualisation, and remote sensing to build a data processing pipeline and a visual application that aid experts and novices alike in labelling Remote Sensing (RS) images.

Convolutional neural networks and graph neural network approaches are shown to encode and model complex RS images successfully. Our approach evolved from using an autoencoder architecture to weakly supervised methods, where we utilise C-means and N-FINDER to generate labels via unsupervised learning. This allowed us to create feature embeddings that represent RS imagery with increasing utility for discriminating between image regions when the learnt feature vectors are reduced to two dimensions in an interactive application.

This is achieved by pseudo-labelling data with clustering or labelling algorithms, see chapters 6 and 7. Given that each pixel can represent a large area, for example, Sentinel 2 has a resolution of 10 meters, a singular label is ineffective in most cases. Therefore, we pseudo-label each pixel with fractional quantities for each class using C-Means (from computer vision) and N-FINDR (from RS). We compare the utility of both methodologies in our work and show that CNNs trained with these labels can, with minimal expansion, achieve performance comparable to state-of-the-art benchmarks.

Representing images with an encoded feature set has been a prominent part of the imaging domain for decades. Standard techniques range from handcrafted to machine learning methods. In chapter 5, we utilised the activation maps from an AEs latent space, which had been designed to be as small as possible, reducing memory and similarity measuring cost. These encodings produced good results for a scene level, or chip; however, they suffered from the orientation of large texture edges outweighing content, negatively affecting the similarity measures. To remove rotational invariance, we introduced methods that use GNNs.

The inclusion of graph neural networks in our image processing pipeline solved rotational invariance and allowed for labelling at a sub-image level, via segmentations. Inherently, due to message passing, GCNs exhibit some degree of neighbourhood encoding. We constructed the initial training graphs to include geographical neighbours, indicated by edges within the graph. This allowed us to embed the spatial autocorrelation of remotely sensed features, as a segment of an image is typically similar to nearby segments rather than to those farther away.

We developed an evolving visual labelling framework. As introduced in chapter 5, we

present each encoding as a two-dimensional embedding, utilising dimensionality-reduction techniques UMAP and t-SNE. This embedding is ultimately how the user interacts with the datasets for labelling which is faster than techniques such as image retrieval. Given the lossy reduction to two dimensions, we also proposed branching features to embed subsets of the data, thereby revealing previously hidden patterns. The user is able to merge labels between the main dataset and subsets, ensuring a smooth workflow. This functionality, introduced with segmentations as described in Chapter 6, enabled our branching methodology to be utilised for labelling finer details. In effect, both an overview and a more detailed approach could be taken, with seamless transition.

8.2 Key Contributions

The contributions of this thesis are the following:

A fully unsupervised labelling tool for exploring large-scale remote sensing datasets: We developed an unsupervised pipeline based on deep convolutional feature representations and manifold learning for exploring and labelling satellite archives. The method learns a generalised embedding space for remote sensing images through a convolutional autoencoder, graph neural networks and manifold learning techniques. The embedding space allows users to efficiently select and label informative samples by exploring a semantically meaningful manifold. The application allows for faster labelling than image retrieval systems, more than threefold, whilst also retaining intra- and inter-class variance between samples.

A novel branching and merging visualisation technique: The application also contained a novel branching and merging functionality. Given the complexity of manifold techniques and long computational times for large datasets, we implement a novel branching technique for sub-manifold refinement. The branching creates a new environment for labelling, refines projected data, and enables expedited labelling. The merging functionality allows users to transfer labels seamlessly between subsets of a dataset and the global view. Further work in the thesis evolved this functionality to include branching and merging as a feature to change efficiently between scene-level image embeddings and segment-level embeddings. Changing how labelling is conducted across different scales within a single interface.

Demonstrated cross-domain generalisation through physical modelling of remote sensing images: We provide empirical evidence that abundance-based pseudo-labelling using N-FINDR provides meaningful generalisation of CNN representations. The pseudo-labels utilised to train a convolutional network for feature representation achieved superior results

to the benchmarked dataset. The convolutional network was not trained on the original data, however, a small subset of a differing dataset.

8.3 Future Work and Discussion

Deep learning has been applied to the analysis of Remote Sensing imagery in various applications, ranging from segmentation to target detection. Despite the advancements in techniques and architectures, many challenges remain. Notably, automated segmentation is needed to reduce reliance on costly expert labelling. Most research in this field focuses on weakly supervised object detection or localisation, where whole images are typically labelled. There are also methods for integrating pixel-level model explainability via activation maps cite rs14215362. In this section, we discuss potential future research avenues and the choices made in this thesis.

The labelling application we have developed can aid the labelling process by utilising the feature encodings of our deep learning models. To project to two dimensions from a high-dimensional feature space, we have utilised dimensionality-reduction techniques from the visualisation literature, namely UMAP and t-SNE. These techniques require the high-dimensional space to be ordered and structured into descriptive patterns so that the resulting projection is as lossless as possible during dimensionality reduction. This process, while effective so far, can be improved by utilising parametric versions of the algorithms. For example, parametric-UMAP [347] utilises neural networks to assist in the embedding. If the user could specify features of interest and steer the reduction phase toward them, this could improve the resulting low-dimensional embedding [240].

Extended work could challenge the embedding spaces constructed by manifold learning projections used for dataset labelling and creation. A key question that remains unanswered is the viability of datasets created from the application. Can a user utilise the application in an efficient manner to find classes of interest with intra- and inter-class variation? What labelling strategies should a user undertake to create the most efficient datasets for downstream classification? These questions would require work to investigate variation modelling in remote sensing images and how variation can be measured. If the variation present via a manifold, presented in this work, is equivalent to the variation required to create a new dataset, if not, what methods can address the limitations? Works could also look to extract variation from the high-dimensional latent space, specifically for dataset building, and methods to visualise that context to the user. In answering these questions, it may be found that not all segments

of the manifold need to be sampled, only at certain intervals based on density, reducing the computational overhead of projecting all samples.

The pipeline presented in this thesis is generalisable to other domains, excluding the works in Chapter 7. The works could apply to many domains where large, unlabelled datasets are the norm. This adaptation could be specific to imaging domains such as medical imaging. Medical imaging and remote sensing both include multispectral images, with large quantities of unlabelled data and inconsistent acquisition conditions. Future works could explore the adaptability of the application for medical applications such as identifying unusual pathologies, forming training sets for segmentation or diagnosis and discovering subtypes of diseases. Any domain that requires accelerated expert annotation could benefit from this thesis.

Remote sensing

Given that RS data inherently represent the Earths surface, the most intuitive way to view the data is through a map, which provides broad context, rather than considering smaller chip-based images. The integration of map-based visualisation of the images could help with understanding the data. However, the balance of views between the two-dimensional embedding space and the map would have to be considered along with interactivity. A simple implementation would have multiple coordinated windows with selection and brushing conducted on both views. Work could be undertaken to include an interactive sliding window on a map projection that provides immediate updates to the embedding space. This immediate update would allow for users to select features on a map, a familiar projection, and directly relate them to the embedding spaces provided.

Within the RS field, spectral indexes are available to create quick assessments of vegetation, water and others [82, 85]. These could be integrated into our pipeline to filter out any chips or segments within our datasets, thereby reducing the number of points projected. For example, if a user wants to filter out all chips that contain water. Finally, in our application, we have only examined scene-level and image segmentation labelling, but other labelling modalities could also be examined, such as bounding boxes for object detection. Methods such as YOLO could be adapted and integrated into our pipeline and application to enable a user to identify similar features within a localised region [8]. Likewise, our application could be expanded to the pixel level or even sub-pixel level, considering abundances measure the fractional proportion of material present within a singular pixel.

With all-level analysis, be it scene, segment, or pixel, work may be undertaken to include on-the-fly improvements to the representations. If the user, whilst exploring the manifolds,

finds a segment or scene misplaced within the manifold, what immediate corrections can be made? For example, can the manifold be temporarily contorted to suit the users intent? This avenue of research would have to be mindful that distorting manifold projections permanently is not beneficial in the long term, the relationship, both globally and locally, is disjointed by explicitly contorting manifolds. However, that also lends to future work in optimising the speed at which the models update the latent space. Given active learning strategies, how many miss-embedded samples and which samples are needed to be labelled in order for an efficient targeted update to the latent space?

Likewise, segments within our work are all produced using a singular algorithm. That algorithm has been proven to provide insufficient segmentation quality, dependent on the feature, for example, long, thin objects. Future work could enable users to identify insufficiently produced segmentation and update the representation by utilising a range of specialised segmentation algorithms in remote sensing [292].

Datasets

For dataset creation, several key considerations should be addressed for any future work. As explored in Chapter 2, any dataset requires consideration of multiple factors, including intra- and inter-class representation. An optimal dataset would consist of the minimum number of samples that contain a wide variation of features and labels. Extracting these labels could be optimised with similarity considerations within our dataset. For example, an airport dataset would require different physical conditions, sizes, building materials, background locations, densities, or numbers of visible planes. In addition, sensing conditions need to be considered, such as seasonality and atmospheric conditions, or for scene-labelled datasets, the location of the airport within the image (which could be partially obscured). Within our high-dimensional feature space representation, a study would need to be undertaken to determine if the manifolds can be sampled in a way that produces samples that vary in every factor [348].

We now address more individual elements of the pipelines utilised to construct the embedding spaces. Starting chronologically from our first technical work in chapter 5 utilising an autoencoder. Utilising variational encodings or a generative adversarial model could bring utility, especially when considering image and label augmentation and perturbations. When considering the downstream need for a smooth transitional space in higher dimensions, the utilisation of such techniques was ideal. In addition, the latent space could be rotationally invariant, depending on the non-linearity between the activation maps and the input to the latent variational space. Likewise, any non-linear neural layers before and after the latent space could

also achieve non-linearity; however, this comes at the cost of complexity. Additionally, variational models are also more suited for generative samples, which could help find more patterns and trends within the low-dimensional space [257]. However, as an initial implementation, we tested the fidelity of the pipeline utilising traditional autoencoders. Then, as we evolved the pipeline to utilise the later layer activation maps, we had to diverge away from using such latent spaces. Firstly, the latent space was not required, and in addition, due to the stochastic nature of the embedding, the activation maps lost fidelity. Potential works could look to include self-supervised learning strategies or masked variants of autoencoders.

Chapter based

In chapter 7 we introduce the N-FINDR algorithm from the remote sensing field to our processing pipeline, and demonstrated its increased utility in comparison to C-means. There are opportunities to include further refinements to our methodology by drawing more techniques from the remote sensing or classical image processing fields and adapting them to our application. For example, the Scale Invariant Feature Transform (SIFT) identifies key points and computes descriptors based on local textures. SIFT is classically computed on greyscale images, and we process multiple activation maps, so the technique and its implementations would need adaptation. We did not approach this within the thesis as the key points are located using areas of intense gradient change; therefore, subtle background features may not be uniformly encoded.

To develop our processing pipeline from its definition in chapter 5, in chapters 6 and 7 we introduce three main concepts: the utilisation of segments, pseudo-labelling and graph networks. There are a large number of algorithms for unsupervised segmentation in the literature for RS images and multiple different methods of optimisation [349]. For our research, we oversegmented using simple linear iterative clustering (SLIC), as the resulting segments provided a balance between cluster homogeneity and uniform area coverage, which aided their representation within the graph neural networks. Other algorithms could provide a better segmentation result, focusing on homogeneity; for example, an entire connected road could be classed as one segment. However, this would also need to be reflected in the downstream graph neural networks as changes in area, shape, and position also need to be taken into consideration when building the graphs. Simply taking the mean of the activation maps may not be enough to discern features.

Pseudo-labelling was initially inspired by endmember extraction and abundances. This field of automated endmember finding has a rich history and numerous avenues for improve-

ment in the literature. As endmembers are extracted as pure pixels, which contain a single material, this process can also be conducted by an expert. Therefore, there could be a generalised framework with automated extraction and expert-guided feature definitions using selected endmembers dependent on their use-case.

We implemented graph networks to encode of neighbourhood information into our feature embedding. Within the graph network, the way each layer learns inherently shares information with its neighbours. Therefore, multiple layers pass messages between the neighbours of a neighbour. An avenue for future research is to examine the degree to which scale and auto-correlation affect the ability of an embedding to discriminate between different physical features. Does local or distance geographic similarity influence the discriminatory skill of a feature embedding? If different features have different optimal layers, we would need to find a way to reflect that into the labelling tools, as different feature spaces would be utilised.

The most computationally expensive algorithm with our pipeline during inference is building the similarity matrix utilising the Hungarian matching algorithm. As each chip is compared to every other in a dataset, the scalability of the pipeline deteriorates. Research could be conducted into encoding entire sub-graph structures [350]. This approach would allow for encoded graph structures to be compared utilising faster similarity measures coded into the visualisation dimensionality reduction, instead of pre-building a similarity matrix.

Bibliography

[1] C. Toth and G. Jóźków, "Remote sensing platforms and sensors: A survey," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 22–36, 2016, theme issue 'State-of-the-art in photogrammetry, remote sensing and spatial information science'. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271615002270

[2] P. Helber, B. Bischke, A. Dengel, and D. Borth, "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*, 2019.

[3] C. Seale, T. Redfern, P. Chatfield, C. Luo, and K. Dempsey, "Coastline detection in satellite imagery: A deep learning approach on new benchmark data," *Remote Sensing of Environment*, vol. 278, p. 113044, 2022.

[4] P. Meixner and F. Leberl, "Vertical- or oblique imagery for semantic building interpretation," in *Vorträge Dreiländertagung OVG, DGPF und SGPF*. DGPF, 2010, pp. 247–256, 2010 Dreiländertagung OVG, DGPF und SGPF ; Conference date: 01-07-2010 Through 03-07-2010.

[5] M. Schmitt, S. A. Ahmadi, and R. Hänsch, "There is no data like more data - current status of machine learning datasets in remote sensing," *CoRR*, vol. abs/2105.11726, 2021. [Online]. Available: https://arxiv.org/abs/2105.11726

[6] F. Gascon, C. Bouzinac, O. Thépaut, M. Jung, B. Francesconi, J. Louis, V. Lonjou, B. Lafrance, S. Massera, A. Gaudel-Vacaresse, F. Languille, B. Alhammoud, F. Viallefont, B. Pflug, J. Bieniarz, S. Clerc, L. Pessiot, T. Trémas, E. Cadau, R. De Bonis, C. Isola, P. Martimort, and V. Fernandez, "Copernicus sentinel-2a

calibration and products validation status," *Remote Sensing*, vol. 9, no. 6, 2017. [Online]. Available: https://www.mdpi.com/2072-4292/9/6/584

[7] A. Sebastianelli, M. P. Del Rosso, and S. L. Ullo, "Automatic dataset builder for machine learning applications to satellite imagery," *SoftwareX*, vol. 15, p. 100739, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711021000728

[8] J. Li, L. Meng, B. Yang, C. Tao, L. Li, and W. Zhang, "Labelrs: An automated toolbox to make deep learning samples from remote sensing images," *Remote Sensing*, vol. 13, no. 11, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/11/2064

[9] N. Li, L. Cheng, L. Wang, H. Chen, Y. Zhang, Y. Yao, J. cheng, and M. Li, "Automatic labelling framework for optical remote sensing object detection samples in a wide area using deep learning," *Expert Systems with Applications*, vol. 255, p. 124827, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424016944

[10] Y. Ma, H. Wu, L. Wang, B. Huang, R. Ranjan, A. Zomaya, and W. Jie, "Remote sensing big data computing: Challenges and opportunities," *Future Generation Computer Systems*, vol. 51, pp. 47–60, 2015, special Section: A Note on New Trends in Data-Aware Scheduling and Resource Provisioning in Modern HPC Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X14002234

[11] F. Bastani and S. Madden, "Beyond road extraction: A dataset for map update using aerial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 905–11 914.

[12] T. Zhang, X. Zhang, J. Li, X. Xu, B. Wang, X. Zhan, Y. Xu, X. Ke, T. Zeng, H. Su, I. Ahmad, D. Pan, C. Liu, Y. Zhou, J. Shi, and S. Wei, "Sar ship detection dataset (ssdd): Official release and comprehensive data analysis," *Remote Sensing*, vol. 13, no. 18, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/18/3690

[13] A. Fuller, K. Millard, and J. R. Green, "Satvit: Pretraining transformers for earth observation," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[14] C. Seale, T. Redfern, P. Chatfield, C. Luo, and K. Dempsey, "Coastline detection in satellite imagery: A deep learning approach on new benchmark data,"

Remote Sensing of Environment, vol. 278, p. 113044, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425722001584

[15] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "Aid: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017.

[16] H. Alemohammad and K. Booth, "Landcovernet: A global benchmark land cover classification training dataset," *arXiv preprint arXiv:2012.03111*, 2020.

[17] W. Zhou, H. Guan, Z. Li, Z. Shao, and M. R. Delavar, "Remote sensing image retrieval in the past decade: Achievements, challenges, and future directions," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 1447–1473, 2023.

[18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[19] Cvat-Ai, "Github - cvat-ai/cvat: Annotate better with cvat, the industry-leading data engine for machine learning. used and trusted by teams at any scale, for data of any scale." [Online]. Available: https://github.com/cvat-ai/cvat

[20] microsoft, "Github - microsoft/vott: Visual object tagging tool: An electron app for building end to end object detection models from images and videos." [Online]. Available: https://github.com/microsoft/VoTT

[21] E. Saralioglu and O. Gungor, "Crowdsourcing in remote sensing: A review of applications and future directions," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 89–110, 2020.

[22] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities," *ACM Comput. Surv.*, vol. 55, no. 13s, Jul. 2023. [Online]. Available: https://doi.org/10.1145/3582688

[23] J. Li, L. Meng, B. Yang, C. Tao, L. Li, and W. Zhang, "Labelrs: An automated toolbox to make deep learning samples from remote sensing images," *Remote Sensing*, vol. 13, no. 11, p. 2064, 2021.

[24] M. Gholizade, H. Soltanizadeh, M. Rahmanimanesh, and S. S. Sana, "A review of recent advances and strategies in transfer learning," *International Journal of System Assurance Engineering and Management*, pp. 1–40, 2025.

[25] A. Pal, "Improving noisy fine-grained datasets using active label cleaning framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7273–7282.

[26] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 4, pp. 213–247, 2022.

[27] S. Wu, J.-M. Cao, and X.-Y. Zhao, "Land cover classification of high-resolution remote sensing images based on improved spectral clustering," *PloS one*, vol. 20, no. 2, p. e0316830, 2025.

[28] J. Du, J. Zelek, D. Zhang, and J. Li, "3dlcdm: Hybrid supervision for land cover discovery mapping of emerging urban structures in 3d remote sensing," *Remote Sensing of Environment*, vol. 331, p. 115018, 2025.

[29] T. Patel, M. W. Jones, and T. Redfern, "Manifold explorer: Satellite image labelling and clustering tool with using deep convolutional autoencoders," *Algorithms*, vol. 16, no. 10, p. 469, 10 2023. [Online]. Available: https://www.mdpi.com/1999-4893/16/10/469

[30] ——, "Leveraging convolutional and graph networks for an unsupervised remote sensing labelling tool," *arXiv preprint arXiv:2508.00506*, Aug. 2025, submitted to Annals of GIS.

[31] N. O. US Department of Commerce and A. Administration, "What is hydrography?" Jun 2013. [Online]. Available: https://oceanservice.noaa.gov/facts/hydrography.html#:~:text=Hydrographic%20surveyors%20study%20these%20bodies,charts%20and%20develop%20hydrographic%20models.

[32] I. H. Office. [Online]. Available: https://iho.int/uploads/user/pubs/standards/s-66/S-66%20Edition%201.1.0_Final_Clean.pdf

[33] I. Davies, "What is a primary charting authority?" Oct 2023. [Online]. Available: https://ukhodigital.blog.gov.uk/2023/10/23/what-is-a-primary-charting-authority/

[34] OceanWise, "Marine environmental monitoring stations (mems) framework for the uk hydrographic office," Jul 2024. [Online]. Available: https://www.oceanwise.eu/marine-environmental-monitoring-stations-mems-framework-for-the-uk-hydrographic-office/

[35] UKHO, Jul 2024. [Online]. Available: https://assets.publishing.service.gov.uk/media/66bcc15c3effd5b79ba490b1/UKHO_AR_2023-24.pdf

[36] D. Stephens, A. Smith, T. Redfern, A. Talbot, A. Lessnoff, and K. Dempsey, "Using three dimensional convolutional neural networks for denoising echosounder point cloud data," *Applied Computing and Geosciences*, vol. 5, p. 100016, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590197419300163

[37] V. Sivaprakasam, D. Lin, M. K. Yetzbacher, H. E. Gemar, J. M. Portier, and A. T. Watnik, "Multi-spectral swir lidar for imaging and spectral discrimination through partial obscurations," *Opt. Express*, vol. 31, no. 4, pp. 5443–5457, Feb 2023. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-31-4-5443

[38] G. A. Shaw and H. K. Burke, "Spectral imaging for remote sensing," *Lincoln laboratory journal*, vol. 14, no. 1, pp. 3–28, 2003.

[39] L. Wang, X. Bai, C. Gong, and F. Zhou, "Hybrid inference network for few-shot sar automatic target recognition," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 11, pp. 9257–9269, 2021.

[40] M. M. Stofa, M. A. Zulkifley, and S. Z. M. Zaki, "A deep learning approach to ship detection using satellite imagery," in *IOP Conference Series: Earth and Environmental Science*, vol. 540, no. 1. IOP Publishing, 2020, p. 012049.

[41] K. Berger, M. Machwitz, M. Kycko, S. C. Kefauver, S. Van Wittenberghe, M. Gerhards, J. Verrelst, C. Atzberger, C. Van der Tol, A. Damm *et al.*, "Multi-sensor spectral synergies for crop stress detection and monitoring in the optical domain: A review," *Remote sensing of environment*, vol. 280, p. 113198, 2022.

[42] H. Jung, H.-S. Choi, and M. Kang, "Boundary enhancement semantic segmentation for building extraction from remote sensed image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2021.

[43] H. Yang, J. Kong, H. Hu, Y. Du, M. Gao, and F. Chen, "A review of remote sensing for water quality retrieval: progress and challenges," *Remote Sensing*, vol. 14, no. 8, p. 1770, 2022.

[44] M. A. Brovelli, Y. Sun, and V. Yordanov, "Monitoring forest change in the amazon using multi-temporal remote sensing data and machine learning classification on google earth engine," *ISPRS International Journal of Geo-Information*, vol. 9, no. 10, p. 580, 2020.

[45] H. Su, Z. Wu, H. Zhang, and Q. Du, "Hyperspectral anomaly detection: A survey," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 1, pp. 64–90, 2022.

[46] T. Hupel and P. Stütz, "Sensor-managed anomaly detection for camouflage detection in airborne multispectral imagery," in *2024 IEEE Aerospace Conference*, 2024, pp. 1–11.

[47] Z. Wu, H. Lu, M. E. Paoletti, H. Su, W. Jing, and J. M. Haut, "Kacnet: Kolmogorov-arnold convolution network for hyperspectral anomaly detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–14, 2025.

[48] A. Kalybekova, "A review of advancements and applications of satellite-derived bathymetry," *Engineered Science*, vol. 35, p. 1541, 2025. [Online]. Available: http://dx.doi.org/10.30919/es1541

[49] M. Ansari, A. Knudby, M. Amani, and M. Sawada, "Retrieving inland water quality parameters via satellite remote sensing: Sensor evaluation, atmospheric correction, and machine learning approaches," *Remote Sensing*, vol. 17, no. 10, 2025. [Online]. Available: https://www.mdpi.com/2072-4292/17/10/1734

[50] J. Chen, S. Chen, R. Fu, D. Li, H. Jiang, C. Wang, Y. Peng, K. Jia, and B. J. Hicks, "Remote sensing big data for water environment monitoring: Current status, challenges, and future prospects," *Earth's Future*, vol. 10, no. 2, p. e2021EF002289, 2022.

[51] J. Dong, T. Zhang, L. Wang, Z. Li, M. Sing Wong, M. Bilal, Z. Zhu, F. Mao, X. Xia, G. Han, Q. Xu, Y. Gu, Y. Lin, B. Zhao, Z. Li, K. Xu, X. Chen, and W. Gong, "First retrieval of daily 160 m aerosol optical depth over urban areas using gaofen-1/6 synergistic observations: Algorithm development and validation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 211, pp. 372–391, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271624001801

[52] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 270–279. [Online]. Available: https://doi.org/10.1145/1869790.1869829

[53] K. Nogueira, J. A. Dos Santos, T. Fornazari, T. S. F. Silva, L. P. Morellato, and R. d. S. Torres, "Towards vegetation species discrimination by using data-driven descriptors," in *2016 9th IAPR Workshop on Pattern Recogniton in Remote Sensing (PRRS)*. Ieee, 2016, pp. 1–6.

[54] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 44–51.

[55] T. X. Guofeng Sheng, Wen Yang and H. Sun, "High-resolution satellite scene classification using a sparse coding based multiple feature combination," *International Journal of Remote Sensing*, vol. 33, no. 8, pp. 2395–2412, 2012. [Online]. Available: https://doi.org/10.1080/01431161.2011.608740

[56] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, pp. 2321–2325, 2015.

[57] B. Zhao, Y. Zhong, G.-S. Xia, and L. Zhang, "Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 4, pp. 2108–2123, 2016.

[58] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271616300144

[59] C. Benedek, X. Descombes, and J. Zerubia, "Building development monitoring in multitemporal remotely sensed image pairs with stochastic birth-death dynamics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 33–50, 2011.

[60] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, "Orientation robust object detection in aerial images using deep convolutional neural network," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 3735–3739.

[61] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, "Accurate object localization in remote sensing images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2486–2498, 2017.

[62] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Contextual classification of lidar data and building object detection in urban areas," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 152–165, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271613002359

[63] W. Zhou, S. Newsam, C. Li, and Z. Shao, "Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, p. 197–209, Nov. 2018. [Online]. Available: http://dx.doi.org/10.1016/j.isprsjprs.2018.01.004

[64] M. Mehmood, A. Shahzad, B. Zafar, A. Shabbir, and N. Ali, "Remote sensing image classification: A comprehensive review and applications," *Mathematical Problems in Engineering*, vol. 2022, no. 1, p. 5880959, 2022.

[65] K. Kikaki, I. Kakogeorgiou, P. Mikeli, D. E. Raitsos, and K. Karantzalos, "Marida: A benchmark for marine debris detection from sentinel-2 remote sensing data," *PLOS ONE*, vol. 17, no. 1, pp. 1–20, 01 2022. [Online]. Available: https://doi.org/10.1371/journal.pone.0262247

[66] I. Gallo, R. La Grassa, N. Landro, and M. Boschetti, "Sentinel 2 time series analysis with 3d feature pyramid network and time domain class activation intervals for crop mapping," *ISPRS International Journal of Geo-Information*, vol. 10, no. 7, 2021. [Online]. Available: https://www.mdpi.com/2220-9964/10/7/483

[67] H. Li, F. Zhu, X. Zheng, M. Liu, and G. Chen, "Mscdunet: A deep learning framework for built-up area change detection integrating multispectral, sar, and vhr data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 5163–5176, 2022.

[68] L. Ding, J. Zhang, H. Guo, K. Zhang, B. Liu, and L. Bruzzone, "Joint spatio-temporal modeling for semantic change detection in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–14, 2024.

[69] A. Berk, P. Conforti, and F. Hawes, "An accelerated line-by-line option for MODTRAN combining on-the-fly generation of line center absorption within 0.1 cm-1 bins and pre-computed line tails," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, M. Velez-Reyes and F. A. Kruse, Eds., vol. 9472, May 2015, p. 947217.

[70] J.-B. Féret, A. Gitelson, S. Noble, and S. Jacquemoud, "Prospect-d: Towards modeling leaf optical properties through a complete lifecycle," *Remote Sensing of Environment*, vol. 193, pp. 204–215, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425717300962

[71] H. Fang, S. Liang, and A. Kuusk, "Retrieving leaf area index using a genetic algorithm with a canopy radiative transfer model," *Remote Sensing of Environment*, vol. 85, no. 3, pp. 257–270, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425703000051

[72] N. Keshava and J. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44–57, 2002.

[73] J. W. Boardman, F. A. Kruse, and R. O. Green, "Mapping target signatures via partial unmixing of aviris data," in *Summaries of the fifth annual JPL airborne earth science workshop. Volume 1: AVIRIS workshop*, 1995.

[74] C.-I. Chang and A. Plaza, "A fast iterative algorithm for implementation of pixel purity index," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 63–67, 2006.

[75] M. E. Winter, "N-findr: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Imaging spectrometry V*, vol. 3753. SPIE, 1999, pp. 266–275.

[76] C.-I. Chang, C.-C. Wu, and C.-T. Tsai, "Random n-finder (n-findr) endmember extraction algorithms for hyperspectral imagery," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 641–656, 2010.

[77] J. Qing, H. Huo, and T. Fang, "Nearest convex hull classifiers for remote sensing classification," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, pp. 589–594, 2008.

[78] F. A. Kruse, A. Lefkoff, y. J. Boardman, K. Heidebrecht, A. Shapiro, P. Barloon, and A. Goetz, "The spectral image processing system (sips)—interactive visualization and analysis of imaging spectrometer data," *Remote sensing of environment*, vol. 44, no. 2-3, pp. 145–163, 1993.

[79] O. A. De Carvalho and P. R. Meneses, "Spectral correlation mapper (scm): an improvement on the spectral angle mapper (sam)," in *Summaries of the 9th JPL Airborne Earth Science Workshop, JPL Publication 00-18*, vol. 9. JPL publication Pasadena, CA, USA, 2000, p. 2.

[80] C.-I. Chang, "An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis," *IEEE Transactions on information theory*, vol. 46, no. 5, pp. 1927–1932, 2000.

[81] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1, pp. 37–52, 1987, proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0169743987800849

[82] J. W. Rouse Jr, R. H. Haas, D. Deering, J. Schell, and J. C. Harlan, "Monitoring the vernal advancement and retrogradation (green wave effect) of natural vegetation," Tech. Rep., 1974.

[83] A. A. Gitelson, O. B. Chivkunova, and M. N. Merzlyak, "Nondestructive estimation of anthocyanins and chlorophylls in anthocyanic leaves," *American Journal of Botany*, vol. 96, no. 10, pp. 1861–1868, 2009.

[84] Z. Jiang, A. R. Huete, K. Didan, and T. Miura, "Development of a two-band enhanced vegetation index without a blue band," *Remote Sensing of Environment*, vol. 112, no. 10, pp. 3833–3845, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425708001971

[85] B. cai Gao, "Ndwi—a normalized difference water index for remote sensing of vegetation liquid water from space," *Remote Sensing of Environment*, vol. 58, no. 3, pp.

257–266, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425796000673

[86] X. Chen, J. E. Vogelmann, M. Rollins, D. Ohlen, C. H. Key, L. Yang, C. Huang, and H. Shi, "Detecting post-fire burn severity and vegetation recovery using multitemporal remote sensing spectral indices and field-collected composite burn index data in a ponderosa pine forest," *International Journal of Remote Sensing*, vol. 32, no. 23, pp. 7905–7927, 2011.

[87] V. V. Salomonson and I. Appel, "Estimating fractional snow cover from modis using the normalized difference snow index," *Remote sensing of environment*, vol. 89, no. 3, pp. 351–360, 2004.

[88] J. D. Braaten, W. B. Cohen, and Z. Yang, "Automated cloud and cloud shadow identification in landsat mss imagery for temperate ecosystems," *Remote Sensing of Environment*, vol. 169, pp. 128–138, 2015.

[89] A. Hollstein, K. Segl, L. Guanter, M. Brell, and M. Enesco, "Ready-to-use methods for the detection of clouds, cirrus, snow, shadow, water and clear sky pixels in sentinel-2 msi images," *Remote Sensing*, vol. 8, no. 8, 2016. [Online]. Available: https://www.mdpi.com/2072-4292/8/8/666

[90] D. K. Hall, G. A. Riggs, and V. V. Salomonson, "Development of methods for mapping global snow cover using moderate resolution imaging spectroradiometer data," *Remote sensing of Environment*, vol. 54, no. 2, pp. 127–140, 1995.

[91] C. A. Cansler and D. McKenzie, "How robust are burn severity indices when applied in a new region? evaluation of alternate field-based and remote-sensing methods," *Remote sensing*, vol. 4, no. 2, pp. 456–483, 2012.

[92] S. Rajendran, F. Sadooni, H. Al-Kuwari, A. Oleg, H. Govil, S. Nasir, and P. Vethamony, "Monitoring oil spill in norilsk, russia using satellite data. sci. rep. 11, 3817," 2021.

[93] J. L. Hatfield and J. H. Prueger, "Value of using different vegetative indices to quantify agricultural crop characteristics at different growth stages under varying management practices," *Remote Sensing*, vol. 2, no. 2, pp. 562–578, 2010.

[94] N. Karasiak, J.-F. Dejoux, M. Fauvel, J. Willm, C. Monteil, and D. Sheeren, "Statistical stability and spatial instability in mapping forest tree species by comparing 9 years of satellite image time series," *Remote Sensing*, vol. 11, no. 21, 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/21/2512

[95] Y. Oyama, T. Fukushima, B. Matsushita, H. Matsuzaki, K. Kamiya, and H. Kobinata, "Monitoring levels of cyanobacterial blooms using the visual cyanobacteria index (vci) and floating algae index (fai)," *International Journal of Applied Earth Observation and Geoinformation*, vol. 38, pp. 335–348, 2015.

[96] E. Evagorou, C. Mettas, A. Agapiou, K. Themistocleous, and D. Hadjimitsis, "Bathymetric maps from multi-temporal analysis of sentinel-2 data: the case study of limassol, cyprus," *Advances in Geosciences*, vol. 45, pp. 397–407, 2019. [Online]. Available: https://adgeo.copernicus.org/articles/45/397/2019/

[97] I. Caballero and R. P. Stumpf, "Retrieval of nearshore bathymetry from sentinel-2a and 2b satellites in south florida coastal waters," *Estuarine, Coastal and Shelf Science*, vol. 226, p. 106277, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0272771418309983

[98] P. Lynch, L. Blesius, and E. Hines, "Classification of urban area using multispectral indices for urban planning," *Remote Sensing*, vol. 12, no. 15, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/15/2503

[99] A. Asokan and J. Anitha, "Machine learning based image processing techniques for satellite image analysis -a survey," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, pp. 119–124.

[100] J. Babbar and N. Rathee, "Satellite image analysis: A review," in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2019, pp. 1–6.

[101] H. Ouchra and A. Belangour, "Satellite image classification methods and techniques: A survey," in *2021 IEEE International Conference on Imaging Systems and Techniques (IST)*, 2021, pp. 1–6.

[102] N. Bagwari, S. Kumar, and V. S. Verma, "A comprehensive review on segmentation techniques for satellite images," *Archives of Computational Methods in Engineering*, vol. 30, no. 7, pp. 4325–4358, 2023.

[103] L. Roberts, "Machine perception of 3-d solids, optical and electro-optical information processing," 1965.

[104] I. E. Sobel, *Camera models and machine perception*. stanford university, 1970.

[105] J. M. Prewitt *et al.*, "Object enhancement and extraction," *Picture processing and Psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970.

[106] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[107] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[108] H. Liu and K. Jezek, "Automated extraction of coastline from satellite imagery by integrating canny edge detection and locally adaptive thresholding methods," *International journal of remote sensing*, vol. 25, no. 5, pp. 937–958, 2004.

[109] L. Shi and Y. Zhao, "Edge detection of high-resolution remote sensing image based on multi-directional improved sobel operator," *IEEE Access*, vol. 11, pp. 135 979–135 993, 2023.

[110] G. Chen, Z. Jiang, and M. Kamruzzaman, "Radar remote sensing image retrieval algorithm based on improved sobel operator," *Journal of Visual Communication and Image Representation*, vol. 71, p. 102720, 2020.

[111] M. Ali and D. Clausi, "Using the canny edge detector for feature extraction and enhancement of remote sensing images," in *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No.01CH37217)*, vol. 5, 2001, pp. 2298–2300 vol.5.

[112] J. Jing, S. Liu, G. Wang, W. Zhang, and C. Sun, "Recent advances on image edge detection: A comprehensive review," *Neurocomputing*, vol. 503, pp. 259–271, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231222008141

[113] Y. Jing, J. An, and Z. Liu, "A novel edge detection algorithm based on global minimization active contour model for oil slick infrared aerial image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2005–2013, 2011.

[114] Y. Gao, M. Hao, Y. Wang, L. Dang, and Y. Guo, "Multi-scale coal fire detection based on an improved active contour model from landsat-8 satellite and uav images," *ISPRS International Journal of Geo-Information*, vol. 10, no. 7, 2021. [Online]. Available: https://www.mdpi.com/2220-9964/10/7/449

[115] X. Wei, W. Zheng, C. Xi, and S. Shang, "Shoreline extraction in sar image based on advanced geometric active contour model," *Remote Sensing*, vol. 13, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/4/642

[116] D. Yang, B. Peng, Z. Al-Huda, A. Malik, and D. Zhai, "An overview of edge and object contour detection," *Neurocomputing*, vol. 488, pp. 470–493, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092523122200248X

[117] A. Sekertekin, "A survey on global thresholding methods for mapping open water body using sentinel-2 satellite imagery and normalized difference water index," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1335–1347, 2021.

[118] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[119] J. M. Prewitt and M. L. Mendelsohn, "The analysis of cell images," *Annals of the New York Academy of Sciences*, vol. 128, no. 3, pp. 1035–1053, 1966.

[120] J. Kapur, P. Sahoo, and A. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 273–285, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0734189X85901252

[121] B. N. Pandey, A. Rana *et al.*, "A literature survey of optimization techniques for satellite image segmentation," in *2018 International conference on advanced computation and telecommunication (ICACAT)*. IEEE, 2018, pp. 1–5.

[122] S. Beucher and F. Meyer, "The morphological approach to segmentation: the watershed transformation," in *Mathematical morphology in image processing*. CRC Press, 2018, pp. 433–481.

[123] I. Rizvi and B. Mohan, "Wavelet based marker-controlled watershed segmentation technique for high resolution satellite images," in *2nd International Conference and workshop on Emerging Trends in Technology (ICWET)*. Citeseer, 2011.

[124] C. Heltin Genitha, M. Sowmya, and T. Sri, "Comparative analysis for the detection of marine vessels from satellite images using fcm and marker-controlled watershed segmentation algorithm," *Journal of the Indian Society of Remote Sensing*, vol. 48, no. 8, pp. 1207–1214, 2020.

[125] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels," 2010.

[126] J. Yang, Y. He, and J. Caspersen, "Region merging using local spectral angle thresholds: A more accurate method for hybrid segmentation of remote sensing images," *Remote Sensing of Environment*, vol. 190, pp. 137–148, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425716304904

[127] L. A. Santos, K. Ferreira, M. Picoli, G. Camara, R. Zurita-Milla, and E.-W. Augustijn, "Identifying spatiotemporal patterns in land use and cover samples from satellite image time series," *Remote Sensing*, vol. 13, no. 5, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/5/974

[128] H. Ji, Z. Zuo, and Q.-L. Han, "A divisive hierarchical clustering approach to hyperspectral band selection," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[129] M. Venkata Dasu, P. Reddy, and S. Chandra Mohan Reddy, "Classification of remote sensing images based on k-means clustering and artificial bee colony optimization," *Advances in Cybernetics, Cognition, and Machine Learning for Communication Technologies*, pp. 57–65, 2020.

[130] B. Kodge, "Extraction and analysis of snow covered area from high resolution satellite imageries using k-means clustering," *Earth Science Informatics*, vol. 16, no. 4, pp. 4285–4291, 2023.

[131] F. H. Wagner, R. Dalagnol, A. H. Sánchez, M. C. Hirye, S. Favrichon, J. H. Lee, S. Mauceri, Y. Yang, and S. Saatchi, "K-textures, a self-supervised hard clustering deep

learning algorithm for satellite image segmentation," *Frontiers in Environmental Science*, vol. 10, p. 946729, 2022.

[132] O. Mezouar, F. Meskine, and I. Boukerch, "Automatic satellite images orthorectification using k–means based cascaded meta-heuristic algorithm," *Photogrammetric Engineering & Remote Sensing*, vol. 89, no. 5, pp. 30–39, 2023.

[133] Y. Zhang, S. Yan, L. Zhang, and B. Du, "Fast projected fuzzy clustering with anchor guidance for multimodal remote sensing imagery," *IEEE Transactions on Image Processing*, vol. 33, pp. 4640–4653, 2024.

[134] X. Han, C. Armenakis, and M. Jadidi, "Dbscan optimization for improving marine trajectory clustering and anomaly detection," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 455–461, 2020.

[135] Y. Cai, Z. Zhang, Z. Cai, X. Liu, X. Jiang, and Q. Yan, "Graph convolutional subspace clustering: A robust subspace clustering framework for hyperspectral image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4191–4202, 2021.

[136] X. Shi, Y. Li, and Q. Zhao, "Flexible hierarchical gaussian mixture model for high-resolution remote sensing image segmentation," *Remote Sensing*, vol. 12, no. 7, p. 1219, 2020.

[137] J. Qu, Q. Du, Y. Li, L. Tian, and H. Xia, "Anomaly detection in hyperspectral imagery based on gaussian mixture model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 11, pp. 9504–9517, 2020.

[138] H. Guan, J. Huang, L. Li, X. Li, S. Miao, W. Su, Y. Ma, Q. Niu, and H. Huang, "Improved gaussian mixture model to map the flooded crops of vv and vh polarization data," *Remote Sensing of Environment*, vol. 295, p. 113714, 2023.

[139] Y. Liu, R. H. Weisberg *et al.*, "A review of self-organizing map applications in meteorology and oceanography," *Self-organizing maps: applications and novel algorithm design*, vol. 1, pp. 253–272, 2011.

[140] F. M. Riese, S. Keller, and S. Hinz, "Supervised and semi-supervised self-organizing maps for regression and classification focusing on hyperspectral data," *Remote Sensing*, vol. 12, no. 1, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/1/7

[141] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[142] A. Vaswani, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[143] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[144] ——, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[145] G. Zhong, L.-N. Wang, X. Ling, and J. Dong, "An overview on data representation learning: From traditional feature learning to recent deep learning," *The Journal of Finance and Data Science*, vol. 2, no. 4, pp. 265–278, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405918816300459

[146] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*. PMLR, 2014, pp. 647–655.

[147] G. Zhong, H. Yao, Y. Liu, C. Hong, and T. Pham, "Classification of photographed document images based on deep-learning features," in *Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*, vol. 10225. SPIE, 2017, pp. 176–181.

[148] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[149] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[150] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, "Attention mechanisms in computer vision: A survey," *Computational visual media*, vol. 8, no. 3, pp. 331–368, 2022.

[151] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 714–735, 1997.

[152] J. Wang, "A survey on graph neural networks," *EAI Endorsed Transactions on e-Learning*, vol. 8, no. 3, pp. e6–e6, 2022.

[153] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[154] M. Meilă and H. Zhang, "Manifold learning: What, how, and why," *Annual Review of Statistics and Its Application*, vol. 11, no. 1, pp. 393–417, 2024.

[155] J. M. Lee, "Smooth manifolds," in *Introduction to smooth manifolds*. Springer, 2003, pp. 1–29.

[156] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[157] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "Multidimensional scaling, sammon mapping, and isomap: Tutorial and survey," *arXiv preprint arXiv:2009.08136*, 2020.

[158] F. Anowar, S. Sadaoui, and B. Selim, "Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne)," *Computer Science Review*, vol. 40, p. 100378, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013721000186

[159] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM journal on scientific computing*, vol. 26, no. 1, pp. 313–338, 2004.

[160] J. Wang, "Laplacian eigenmaps," in *Geometric structure of high-dimensional data and dimensionality reduction*. Springer, 2012, pp. 235–247.

[161] L. van der Maaten, "Barnes-hut-sne," 2013.

[162] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2020.

[163] J. Cook, I. Sutskever, A. Mnih, and G. Hinton, "Visualizing similarity data with a mixture of maps," in *Artificial intelligence and statistics*. PMLR, 2007, pp. 67–74.

[164] N. Pettorelli, *The normalized difference vegetation index*. Oxford University Press, USA, 2013.

[165] M. Main-Knorn, B. Pflug, J. Louis, V. Debaecker, U. Müller-Wilm, and F. Gascon, "Sen2cor for sentinel-2," in *Image and Signal Processing for Remote Sensing XXIII*, vol. 10427. SPIE, 2017, pp. 37–48.

[166] M. A. Günen and U. H. Atasever, "Remote sensing and monitoring of water resources: A comparative study of different indices and thresholding methods," *Science of the Total Environment*, vol. 926, p. 172117, 2024.

[167] Y. Zhang, Q. Lin, L. Li, Z. Xiao, Z. Ming, and V. C. Leung, "Multiobjective band selection approach via an adaptive particle swarm optimizer for remote sensing hyperspectral images," *Swarm and Evolutionary Computation*, vol. 89, p. 101614, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650224001524

[168] S. Hu, F. Gao, X. Zhou, J. Dong, and Q. Du, "Hybrid convolutional and attention network for hyperspectral image denoising," *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, 2024.

[169] H. Zhai, H. Zhang, L. Zhang, and P. Li, "Cloud/shadow detection based on spectral indices for multi/hyperspectral optical remote sensing imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 144, pp. 235–253, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271618301989

[170] S. Qiu, Z. Zhu, and B. He, "Fmask 4.0: Improved cloud and cloud shadow detection in landsats 4–8 and sentinel-2 imagery," *Remote Sensing of Environment*, vol. 231, p. 111205, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425719302172

[171] Y. Guo, X. Cao, B. Liu, and M. Gao, "Cloud detection for satellite imagery using attention-based u-net convolutional neural network," *Symmetry*, vol. 12, no. 6, 2020. [Online]. Available: https://www.mdpi.com/2073-8994/12/6/1056

[172] M. Segal-Rozenhaimer, A. Li, K. Das, and V. Chirayath, "Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (cnn)," *Remote Sensing of Environment*, vol. 237, p. 111446, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425719304651

[173] L. Sun, X. Yang, S. Jia, C. Jia, Q. Wang, X. Liu, J. Wei, and X. Zhou, "Satellite data cloud detection using deep learning supported by hyperspectral data," *International Journal of Remote Sensing*, vol. 41, no. 4, pp. 1349–1371, 2020.

[174] X. Wen, Z. Pan, Y. Hu, and J. Liu, "Generative adversarial learning in yuv color space for thin cloud removal on satellite imagery," *Remote Sensing*, vol. 13, no. 6, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/6/1079

[175] J. Zheng, X.-Y. Liu, and X. Wang, "Single image cloud removal using u-net and generative adversarial networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 6371–6385, 2021.

[176] H. Pan, "Cloud removal for remote sensing imagery via spatial attention generative adversarial network," 2020. [Online]. Available: https://arxiv.org/abs/2009.13015

[177] F. N. Darbaghshahi, M. R. Mohammadi, and M. Soryani, "Cloud removal in remote sensing images using generative adversarial networks and sar-to-optical image translation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–9, 2022.

[178] P. Ebel, V. S. F. Garnot, M. Schmitt, J. D. Wegner, and X. X. Zhu, "Uncrtaints: Uncertainty quantification for cloud removal in optical satellite time series," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023, pp. 2086–2096.

[179] M. Schott, A. Zell, S. Lautenbach, G. Sumbul, M. Schultz, A. Zipf, and B. Demir, *Analyzing and Improving the Quality and Fitness for Purpose of OpenStreetMap as Labels in Remote Sensing Applications*. Cham: Springer Nature Switzerland, 2024, pp. 21–42. [Online]. Available: https://doi.org/10.1007/978-3-031-35374-1_2

[180] A. Babenko and V. Lempitsky, "Aggregating deep convolutional features for image retrieval," *arXiv preprint arXiv:1510.07493*, 2015.

[181] F. Radenović, G. Tolias, and O. Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 3–20.

[182] F. Jiang, H.-M. Hu, J. Zheng, and B. Li, "A hierarchal bow for image retrieval by enhancing feature salience," *Neurocomputing*, vol. 175, pp. 146–154, 2016.

[183] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311.

[184] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[185] A. Jimenez, J. M. Alvarez, and X. Giro-i Nieto, "Class-weighted convolutional features for visual instance search," *arXiv preprint arXiv:1707.02581*, 2017.

[186] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 685–701.

[187] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *arXiv preprint arXiv:1511.05879*, 2015.

[188] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an lstm-based saliency attentive model," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5142–5154, 2018.

[189] V. Rupapara, M. Narra, N. K. Gonda, K. Thipparthy, and S. Gandhi, "Auto-encoders for content-based image retrieval with its implementation using handwritten dataset," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020, pp. 289–294.

[190] A. E. Minarno, K. M. Ghufron, T. S. Sabrila, L. Husniah, and F. D. S. Sumadi, "Cnn based autoencoder application in breast cancer image retrieval," in *2021 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2021, pp. 29–34.

[191] Y. Liang and W. Liang, "Reswcae: Biometric pattern image denoising using residual wavelet-conditioned autoencoder," *arXiv preprint arXiv:2307.12255*, 2023.

[192] S. R. Singh, S. R. Dubey, S. MS, S. Ventrapragada, and S. S. Dasharatha, "Joint triplet autoencoder for histopathological colon cancer nuclei retrieval," *Multimedia Tools and Applications*, vol. 83, no. 1, pp. 1063–1082, 2024.

[193] K. Zhang, S. Qi, J. Cai, D. Zhao, T. Yu, Y. Yue, Y. Yao, and W. Qian, "Content-based image retrieval with a convolutional siamese neural network: Distinguishing lung cancer and tuberculosis in ct images," *Computers in biology and medicine*, vol. 140, p. 105096, 2022.

[194] Ş. Öztürk, "Hash code generation using deep feature selection guided siamese network for content-based medical image retrieval," *Gazi University Journal of Science*, vol. 34, no. 3, pp. 733–746, 2021.

[195] G. V. R. M. Kumar and D. Madhavi, "Stacked siamese neural network (ssinn) on neural codes for content-based image retrieval," *IEEE Access*, vol. 11, pp. 77 452–77 463, 2023.

[196] Y. Liu, L. Ding, C. Chen, and Y. Liu, "Similarity-based unsupervised deep transfer learning for remote sensing image retrieval," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 11, pp. 7872–7889, 2020.

[197] Y. Li, J. Ma, and Y. Zhang, "Image retrieval from remote sensing big data: A survey," *Information Fusion*, vol. 67, pp. 94–115, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253520303778

[198] Z. Yuan, W. Zhang, C. Tian, X. Rong, Z. Zhang, H. Wang, K. Fu, and X. Sun, "Remote sensing cross-modal text-image retrieval based on global and local information," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[199] C. Liu, J. Ma, X. Tang, F. Liu, X. Zhang, and L. Jiao, "Deep hash learning for remote sensing image retrieval," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 4, pp. 3420–3443, 2021.

[200] M. Huang, L. Dong, W. Dong, and G. Shi, "Supervised contrastive learning based on fusion of global and local features for remote sensing image retrieval," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–13, 2023.

[201] Y. Zhang, X. Zheng, and X. Lu, "Remote sensing image retrieval by deep attention hashing with distance-adaptive ranking," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 4301–4311, 2023.

[202] D. Zhao, Y. Chen, and S. Xiong, "Multiscale context deep hashing for remote sensing image retrieval," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 7163–7172, 2023.

[203] L. Han, M. E. Paoletti, X. Tao, Z. Wu, J. M. Haut, J. Plaza, and A. Plaza, "Central cohesion gradual hashing for remote sensing image retrieval," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.

[204] J. Regan and M. Khodayar, "A triplet graph convolutional network with attention and similarity-driven dictionary learning for remote sensing image retrieval," *Expert Systems with Applications*, vol. 232, p. 120579, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423010813

[205] G. Sumbul, M. Ravanbakhsh, and B. Demir, "Informative and representative triplet selection for multilabel remote sensing image retrieval," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[206] J. Henkel, G. Hoxha, G. Sumbul, L. Möllenbrok, and B. Demir, "Annotation cost efficient active learning for content based image retrieval," in *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, 2023, pp. 4994–4997.

[207] A. Shabbir, N. Ali, J. Ahmed, B. Zafar, A. Rasheed, M. Sajid, A. Ahmed, and S. H. Dar, "Satellite and scene image classification based on transfer learning and fine tuning of resnet50," *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 5843816, 2021.

[208] W. Zhou, H. Guan, Z. Li, Z. Shao, and M. R. Delavar, "Remote sensing image retrieval in the past decade: Achievements, challenges, and future directions," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 1447–1473, 2023.

[209] R. Nara, Y.-C. Lin, Y. Nozawa, Y. Ng, G. Itoh, O. Torii, and Y. Matsui, "Revisiting relevance feedback for clip-based interactive image retrieval," in *Computer Vision – ECCV 2024 Workshops*, A. Del Bue, C. Canton, J. Pont-Tuset, and T. Tommasi, Eds. Cham: Springer Nature Switzerland, 2025, pp. 1–16.

[210] L. Vadicamo, F. Scotti, A. Dearle, and R. Connor, "Comparative analysis of relevance feedback techniques for image retrieval," in *International Conference on Multimedia Modeling*. Springer, 2025, pp. 206–219.

[211] R. Younas, H. B. U. Haq, and M. D. Baig, "A framework for extensive content-based image retrieval system incorporating relevance feedback and query suggestion," *Spectrum of Operational Research*, vol. 1, no. 1, pp. 13–32, 2024.

[212] Z. Gui, X. Liu, A. Zhao, Y. Jiang, Z. Ling, X. Hu, F. Li, Z. Yang, H. Wu, and S. Zhao, "Map retrieval intention recognition based on relevance feedback and geographic semantic guidance: For better understanding user retrieval demands," *Information Processing & Management*, vol. 61, no. 4, p. 103767, 2024.

[213] S. Kumar, A. Jain, S. Rani, D. Ghai, S. Achampeta, and P. Raja, "Enhanced sbir based re-ranking and relevance feedback," in *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*. IEEE, 2021, pp. 7–12.

[214] O. Ghozatlou, M. Datcu, A. Focsa, M. Heredia Conde, and S. L. Ullo, "A review and a perspective of deep active learning for remote sensing image analysis: Enhanced adaptation to user conjecture," *IEEE Geoscience and Remote Sensing Magazine*, vol. 12, no. 3, pp. 125–148, 2024.

[215] G. Lenczner, A. Chan-Hon-Tong, B. Le Saux, N. Luminari, and G. Le Besnerais, "Dial: Deep interactive and active learning for semantic segmentation in remote sensing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 3376–3389, 2022.

[216] L. Möllenbrok, G. Sumbul, and B. Demir, "Deep active learning for multi-label classification of remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.

[217] I. Kalita, R. N. Sai Kumar, and M. Roy, "Deep learning-based cross-sensor domain adaptation under active learning for land cover classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[218] K. S. Miller and A. L. Bertozzi, "Model change active learning in graph-based semi-supervised learning," *Communications on Applied Mathematics and Computation*, vol. 6, no. 2, pp. 1270–1298, 2024.

[219] X. Di, Z. Xue, and M. Zhang, "Active learning-driven siamese network for hyperspectral image classification," *Remote Sensing*, vol. 15, no. 3, p. 752, 2023.

[220] L. Eversberg and J. Lambrecht, "Combining synthetic images and deep active learning: Data-efficient training of an industrial object detection model," *Journal of Imaging*, vol. 10, no. 1, 2024.

[221] H. Hong, S. Yan, S. Feng, Y. Yan, and Y. Hong, "Galot: Generative active learning via optimizable zero-shot text-to-image generation," *arXiv preprint arXiv:2412.16227*, 2024.

[222] X. Wang, L. Sun, A. Chehri, and Y. Song, "A review of gan-based super-resolution reconstruction for optical remote sensing images," *Remote Sensing*, vol. 15, no. 20, p. 5062, 2023.

[223] X. Feng, W. Zhang, X. Su, and Z. Xu, "Optical remote sensing image denoising and super-resolution reconstructing using optimized generative network in wavelet transform domain," *Remote Sensing*, vol. 13, no. 9, p. 1858, 2021.

[224] A. Singh and L. Bruzzone, "Sigan: Spectral index generative adversarial network for data augmentation in multispectral remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[225] Z. Yu, C. Liu, L. Liu, Z. Shi, and Z. Zou, "Metaearth: A generative foundation model for global-scale remote sensing image generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 3, pp. 1764–1781, 2025.

[226] G. Wang and P. Ren, "Hyperspectral image classification with feature-oriented adversarial active learning," *Remote Sensing*, vol. 12, no. 23, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/23/3879

[227] B. D. Casey Baker, E. Immel, and C. Bogart, "Object detection on streaming lidar data with active learning," *exptechinc*, 2025.

[228] M. Kölle, V. Walter, and U. Sörgel, "Building a fully-automatized active learning framework for the semantic segmentation of geospatial 3d point clouds," *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 92, no. 2, pp. 131–161, 2024.

[229] M. Schmitt, L. H. Hughes, C. Qiu, and X. X. Zhu, "Sen12ms–a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion," *arXiv preprint arXiv:1906.07789*, 2019.

[230] F. Bastani, P. Wolters, R. Gupta, J. Ferdinando, and A. Kembhavi, "Satlaspretrain: A large-scale dataset for remote sensing image understanding," 2023. [Online]. Available: https://arxiv.org/abs/2211.15660

[231] E. Dritsas and M. Trigka, "Remote sensing and geospatial analysis in the big data era: A survey," *Remote Sensing*, vol. 17, no. 3, 2025. [Online]. Available: https://www.mdpi.com/2072-4292/17/3/550

[232] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE symposium on visual languages*. IEEE, 1996, pp. 336–343.

[233] B. Shneiderman, C. Plaisant, M. Cohen, S. M. Jacobs, and N. Elmqvist, *Designing the user interface: Strategies for effective human-computer interaction*. Pearson, 2018.

[234] C. Ware, *Information visualization: perception for design*. Morgan Kaufmann, 2019.

[235] M. Neves and J. Ševa, "An extensive review of tools for manual annotation of documents," *Briefings in bioinformatics*, vol. 22, no. 1, pp. 146–163, 2021.

[236] K. Hu, S. Gaikwad, M. Hulsebos, M. A. Bakker, E. Zgraggen, C. Hidalgo, T. Kraska, G. Li, A. Satyanarayan, and Ç. Demiralp, "Viznet: Towards a large-scale visualization learning and benchmarking repository," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–12.

[237] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "Visual interaction with dimensionality reduction: A structured literature analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 241–250, 2017.

[238] L. Bradel, C. North, and L. House, "Multi-model semantic interaction for text analytics," in *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2014, pp. 163–172.

[239] C. Heine and G. Scheuermann, "Manual clustering refinement using interaction with blobs," in *Proceedings of the 9th Joint Eurographics/IEEE VGTC conference on Visualization*, 2007, pp. 59–66.

[240] N. Pezzotti, B. P. F. Lelieveldt, L. v. d. Maaten, T. Höllt, E. Eisemann, and A. Vilanova, "Approximated and user steerable tsne for progressive visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1739–1752, 2017.

[241] M. Ali, M. W. Jones, X. Xie, and M. Williams, "Timecluster: dimension reduction applied to temporal data for visual analytics," *The Visual Computer*, vol. 35, no. 6-8, pp. 1013–1026, 2019.

[242] M. Ali, R. Borgo, and M. W. Jones, "Concurrent time-series selections using deep learning and dimension reduction," *Knowledge-Based Systems*, vol. 233, p. 107507.

[243] A. Alqahtani, M. Ali, X. Xie, and M. W. Jones, "Deep time-series clustering: A review," *Electronics*, vol. 10, no. 23, p. 3001.

[244] M. Boschetti, F. Nutini, G. Manfron, P. A. Brivio, and A. Nelson, "Comparative analysis of normalised difference spectral indices derived from modis for detecting surface water in flooded rice cropping systems," *PloS one*, vol. 9, no. 2, p. e88741, 2014.

[245] L. Zhou, L. Tsang, V. Jandhyala, and C.-T. Chen, "Studies on accuracy of numerical simulations of emission from rough ocean-like surfaces," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 8, pp. 1757–1763, 2001.

[246] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS journal of photogrammetry and remote sensing*, vol. 66, no. 3, pp. 247–259, 2011.

[247] C. Wemmert, A. Puissant, G. Forestier, and P. Gancarski, "Multiresolution remote sensing image clustering," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 3, pp. 533–537, 2009.

[248] C. Mu, Y. Liu, and Y. Liu, "Hyperspectral image spectral–spatial classification method based on deep adaptive feature fusion," *Remote Sensing*, vol. 13, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/4/746

[249] Q. Zhu, Y. Zhong, Y. Liu, L. Zhang, and D. Li, "A deep-local-global feature fusion framework for high spatial resolution imagery scene classification," *Remote Sensing*, vol. 10, no. 4, 2018. [Online]. Available: https://www.mdpi.com/2072-4292/10/4/568

[250] S. Bakheet, A. Al-Hamadi, E. Soliman, and M. Heshmat, "Hybrid bag-of-visual-words and featurewiz selection for content-based visual information retrieval," *Sensors*, vol. 23, no. 3, p. 1653, 2023.

[251] X.-Y. Tong, G.-S. Xia, F. Hu, Y. Zhong, M. Datcu, and L. Zhang, "Exploiting deep features for remote sensing image retrieval: A systematic investigation," *IEEE Transactions on Big Data*, vol. 6, no. 3, pp. 507–521, 2020.

[252] E. Kordi Ghasrodashti and N. Sharma, "Hyperspectral image classification using an extended auto-encoder method," *Signal Processing: Image Communication*, vol. 92, p. 116111, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596520302290

[253] E. Protopapadakis, A. Doulamis, N. Doulamis, and E. Maltezos, "Stacked autoencoders driven by semi-supervised learning for building extraction from near infrared remote sensing imagery," *Remote Sensing*, vol. 13, no. 3, p. 371, 2021.

[254] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS journal of photogrammetry and remote sensing*, vol. 152, pp. 166–177, 2019.

[255] L. Windrim, R. Ramakrishnan, A. Melkumyan, R. J. Murphy, and A. Chlingaryan, "Unsupervised feature-learning for hyperspectral data with autoencoders," *Remote Sensing*, vol. 11, no. 7, p. 864, 2019.

[256] L. T. Luppino, M. A. Hansen, M. Kampffmeyer, F. M. Bianchi, G. Moser, R. Jenssen, and S. N. Anfinsen, "Code-aligned autoencoders for unsupervised change detection in multimodal remote sensing images," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[257] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and their applications in machine learning: a survey," *Artificial intelligence review*, vol. 57, no. 2, p. 28, 2024.

[258] X. Shen, B. Liu, Y. Zhou, J. Zhao, and M. Liu, "Remote sensing image captioning via variational autoencoder and reinforcement learning," *Knowledge-Based Systems*, vol. 203, p. 105920, 2020.

[259] Y. Zerrouki, F. Harrou, N. Zerrouki, A. Dairi, and Y. Sun, "Desertification detection using an improved variational autoencoder-based approach through etm-landsat satellite data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 202–213, 2020.

[260] P. Naik, M. Dalponte, and L. Bruzzone, "A disentangled variational autoencoder for prediction of above ground biomass from hyperspectral data," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 2021, pp. 2991–2994.

[261] J. Lou, B. Liu, Y. Xiong, X. Zhang, and X. Yuan, "Variational autoencoder framework for hyperspectral retrievals (hyper-vae) of phytoplankton absorption and chlorophyll a in coastal waters for nasa's emit and pace missions," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–16, 2025.

[262] Y. Cong, S. Khanna, C. Meng, P. Liu, E. Rozi, Y. He, M. Burke, D. Lobell, and S. Ermon, "Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 197–211. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/01c561df365429f33fcd7a7faa44c985-Paper-Conference.pdf

[263] S. Girtsou, E. D. Salas-Porras, L. Freischem, J. Massant, K.-M. Bintsi, G. Castiglione, W. Jones, M. Eisinger, E. Johnson, and A. Jungbluth, "3d cloud reconstruction through geospatially-aware masked autoencoders," *arXiv preprint arXiv:2501.02035*, 2025.

[264] H. Yan, S. Su, M. Wu, M. Xu, Y. Zuo, C. Zhang, and B. Huang, "Seamae: Masked pre-training with meteorological satellite imagery for sea fog detection," *Remote Sensing*, vol. 15, no. 16, p. 4102, 2023.

[265] Z. Wang, L. Zhao, and W. Xing, "Stylediffusion: Controllable disentangled style transfer via diffusion models," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 7643–7655.

[266] K. Ayush, B. Uzkent, C. Meng, K. Tanmay, M. Burke, D. Lobell, and S. Ermon, "Geography-aware self-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 181–10 190.

[267] Z. Hu, K. Gao, J. Wang, Z. Yang, Z. Zhang, H. Cheng, and W. Li, "Enhanced grounding dino: Efficient cross-modality block for open-set object detection in remote sensing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 15 291–15 303, 2025.

[268] R. Vaddi, B. Phaneendra Kumar, P. Manoharan, L. Agilandeeswari, and V. Sangeetha, "Strategies for dimensionality reduction in hyperspectral remote sensing: A comprehensive overview," *The Egyptian Journal of Remote Sensing and Space Sciences*, vol. 27, no. 1, pp. 82–92, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S111098232400005X

[269] E. Martel, R. Lazcano, J. López, D. Madroñal, R. Salvador, S. López, E. Juarez, R. Guerra, C. Sanz, and R. Sarmiento, "Implementation of the principal component analysis onto high-performance computer facilities for hyperspectral dimensionality reduction: Results and comparisons," *Remote Sensing*, vol. 10, no. 6, 2018. [Online]. Available: https://www.mdpi.com/2072-4292/10/6/864

[270] Y. Wang, H. H. Hernández, C. M. Albrecht, and X. X. Zhu, "Feature guided masked autoencoder for self-supervised learning in remote sensing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 321–336, 2025.

[271] P. Zhu, "An empirical comparative study of classical dimensionality reduction methods: Mds, isomap, and lle," *preprint*, 2025.

[272] S. Lee and S. Choi, "Landmark mds ensemble," *Pattern Recognition*, vol. 42, no. 9, pp. 2045–2053, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320308005049

[273] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[274] C. J. Dsilva, R. Talmon, R. R. Coifman, and I. G. Kevrekidis, "Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case

study," *Applied and Computational Harmonic Analysis*, vol. 44, no. 3, pp. 759–773, 2018.

[275] S. K. McFEETERS, "The use of the normalized difference water index (ndwi) in the delineation of open water features," *International Journal of Remote Sensing*, vol. 17, no. 7, pp. 1425–1432, 1996. [Online]. Available: https://doi.org/10.1080/01431169608948714

[276] M. Sedlmair, T. Munzner, and M. Tory, "Empirical guidance on scatterplot and dimension reduction technique choices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2634–2643, 2013.

[277] D. Kobak and G. C. Linderman, "Umap does not preserve global structure any better than t-sne when using the same initialization," *BioRxiv*, pp. 2019–12, 2019.

[278] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE Symposium on Visual Languages*, 1996, pp. 336–343.

[279] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," 2018.

[280] Y. Liu, D. Chen, S. Fu, P. T. Mathiopoulos, M. Sui, J. Na, and J. Peethambaran, "Segmentation of individual tree points by combining marker-controlled watershed segmentation and spectral clustering optimization," *Remote Sensing*, vol. 16, no. 4, p. 610, 2024.

[281] X. Tang, X. Huang, Z. Xiong, X. Wang, and Z. Zhan, "An adaptive superpixels for vegetation detection on high resolution images based on mlp," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 187–195, 2022.

[282] Y. Liu, D. Chen, S. Fu, P. T. Mathiopoulos, M. Sui, J. Na, and J. Peethambaran, "Segmentation of individual tree points by combining marker-controlled watershed segmentation and spectral clustering optimization," *Remote Sensing*, vol. 16, no. 4, 2024. [Online]. Available: https://www.mdpi.com/2072-4292/16/4/610

[283] M. Zhang, Y. Ge, Y. Xue, and J. Zhao, "Identification of geomorphological hazards in an underground coal mining area based on an improved region merging watershed algorithm," *Arabian Journal of Geosciences*, vol. 13, no. 9, p. 339, 2020.

[284] J. Yu, X. He, P. Yang, M. Motagh, J. Xu, and J. Xiong, "Coastal aquaculture extraction using gf-3 fully polarimetric sar imagery: A framework integrating unet++ with marker-controlled watershed segmentation," *Remote Sensing*, vol. 15, no. 9, p. 2246, 2023.

[285] C. Li and M. G. M. Johar, "Urban built-up area recognition in remote sensing images using deep learning," in *2025 5th International Symposium on Computer Technology and Information Science (ISCTIS)*, 2025, pp. 233–237.

[286] F. Wang, X. Du, W. Zhang, L. Nie, H. Wang, S. Zhou, and J. Ma, "Remote sensing lidar and hyperspectral classification with multi-scale graph encoder–decoder network," *Remote Sensing*, vol. 16, no. 20, p. 3912, 2024.

[287] R. Guan, W. Tu, S. Wang, J. Liu, D. Hu, C. Tang, Y. Feng, J. Li, B. Xiao, and X. Liu, "Structure-adaptive multi-view graph clustering for remote sensing data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 16, 2025, pp. 16 933–16 941.

[288] K. Wang, Y. Men, Y. Liu, J. Li, R. Zhao, and C. Men, "Superpixel segmentation of remote sensing images via edge extension and adaptive region merging," *Journal of Visual Communication and Image Representation*, vol. 113, p. 104628, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320325002421

[289] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *CVPR 2011*, 2011, pp. 2097–2104.

[290] ——, "Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 99–112, 2013.

[291] H. Zhao, F. Zhou, L. Bruzzone, R. Guan, and C. Yang, "Superpixel-level global and local similarity graph-based clustering for large hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[292] P. J and B. V. Kumar, "An extensive survey on superpixel segmentation: A research perspective," *Archives of Computational Methods in Engineering*, vol. 30, no. 6, pp. 3749–3767, 07 2023, copyright - © The Author(s) under exclusive licence to

International Center for Numerical Methods in Engineering (CIMNE) 2023; Last updated - 2025-10-03. [Online]. Available: https://www.proquest.com/scholarly-journals/extensive-survey-on-superpixel-segmentation/docview/3256681669/se-2

[293] M. P. Barbato, P. Napoletano, F. Piccoli, and R. Schettini, "Unsupervised segmentation of hyperspectral remote sensing images with superpixels," *Remote Sensing Applications: Society and Environment*, vol. 28, p. 100823, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352938522001318

[294] B. Sasmal and K. G. Dhal, "A survey on the utilization of superpixel image for clustering based image segmentation," *Multimedia Tools and Applications*, vol. 82, no. 23, pp. 35 493–35 555, 2023.

[295] X. Zhao, J. Ma, L. Wang, Z. Zhang, Y. Ding, and X. Xiao, "A review of hyperspectral image classification based on graph neural networks," *Artificial Intelligence Review*, vol. 58, no. 6, p. 172, 2025.

[296] P. Sellars, A. I. Aviles-Rivero, and C.-B. Schönlieb, "Superpixel contracted graph-based learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 6, pp. 4180–4193, 2020.

[297] C. Zhao, B. Qin, S. Feng, W. Zhu, W. Sun, W. Li, and X. Jia, "Hyperspectral image classification with multi-attention transformer and adaptive superpixel segmentation-based active learning," *IEEE Transactions on Image Processing*, vol. 32, pp. 3606–3621, 2023.

[298] Q. Diao, Y. Dai, J. Wang, X. Feng, F. Pan, and C. Zhang, "Spatial-pooling-based graph attention u-net for hyperspectral image classification," *Remote Sensing*, vol. 16, no. 6, 2024. [Online]. Available: https://www.mdpi.com/2072-4292/16/6/937

[299] K. Wu, Y. Zhan, Y. An, and S. Li, "Multiscale feature search-based graph convolutional network for hyperspectral image classification," *Remote Sensing*, vol. 16, no. 13, 2024. [Online]. Available: https://www.mdpi.com/2072-4292/16/13/2328

[300] Y. Ding, Z. Zhang, X. Zhao, D. Hong, W. Cai, N. Yang, and B. Wang, "Multi-scale receptive fields: Graph attention neural network for hyperspectral image classification," *Expert Systems with Applications*, vol. 223, p. 119858, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423003597

[301] Q. Diao, Y. Dai, C. Zhang, Y. Wu, X. Feng, and F. Pan, "Superpixel-based attention graph neural network for semantic segmentation in aerial images," *Remote Sensing*, vol. 14, no. 2, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/2/305

[302] Y. Zhao and F. Yan, "Hyperspectral image classification based on sparse superpixel graph," *Remote Sensing*, vol. 13, no. 18, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/18/3592

[303] M. S. Kotzagiannidis and C.-B. Schönlieb, "Semi-supervised superpixel-based multi-feature graph learning for hyperspectral image data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[304] H. Zeng, Q. Liu, M. Zhang, X. Han, and Y. Wang, "Semi-supervised hyperspectral image classification with graph clustering convolutional networks," *arXiv preprint arXiv:2012.10932*, 2020.

[305] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[306] Y. Zhang, S. Yan, L. Zhang, and B. Du, "Fast projected fuzzy clustering with anchor guidance for multimodal remote sensing imagery," *IEEE Transactions on Image Processing*, 2024.

[307] S. A. Taghanaki, Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, and G. Hamarneh, "Combo loss: Handling input and output imbalance in multi-organ segmentation," *Computerized Medical Imaging and Graphics*, vol. 75, pp. 24–33, 2019.

[308] Q. Diao, Y. Dai, C. Zhang, Y. Wu, X. Feng, and F. Pan, "Superpixel-based attention graph neural network for semantic segmentation in aerial images," *Remote Sensing*, vol. 14, no. 2, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/2/305

[309] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021.

[310] H. W. Kuhn, *The Hungarian Method for the Assignment Problem*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 29–47. [Online]. Available: https://doi.org/10.1007/978-3-540-68279-0_2

[311] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2020. [Online]. Available: https://arxiv.org/abs/1802.03426

[312] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[313] D. chen He and L. Wang, "Texture unit, texture spectrum, and texture analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, no. 4, pp. 509–512, 1990.

[314] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[315] F. Kruse, A. Lefkoff, J. Boardman, K. Heidebrecht, A. Shapiro, P. Barloon, and A. Goetz, "The spectral image processing system (sips)—interactive visualization and analysis of imaging spectrometer data," *Remote Sensing of Environment*, vol. 44, no. 2, pp. 145–163, 1993, airbone Imaging Spectrometry. [Online]. Available: https://www.sciencedirect.com/science/article/pii/003442579390013N

[316] N. Iqbal, R. Mumtaz, U. Shafi, and S. M. H. Zaidi, "Gray level co-occurrence matrix (glcm) texture based crop classification using low altitude remote sensing platforms," *PeerJ Computer Science*, vol. 7, p. e536, 2021.

[317] Y. Liu and J. Zhang, "Deep and shallow feature fusion framework for remote sensing open pit coal mine scene recognition," *Scientific Reports*, vol. 14, no. 1, p. 24124, 2024.

[318] A. Ye, X. Zhou, Y. Gong, F. Miao, and H. Zhao, "Sample labeling and classification method of hyperspectral remote sensing images based on texture features and semi-supervised learning," *Geoscientific Instrumentation, Methods and Data Systems Discussions*, vol. 2023, pp. 1–17, 2023.

[319] A. Kiruluta, E. Lundy, and A. Lemos, "Novel change detection framework in remote sensing imagery using diffusion models and structural similarity index (ssim)," *arXiv preprint arXiv:2408.10619*, 2024.

[320] T. Hu, P. Gao, S. Ye, and S. Shen, "Improved sr-ssim band selection method based on band subspace partition," *Remote Sensing*, vol. 15, no. 14, p. 3596, 2023.

[321] J.-F. Pambrun and R. Noumeir, "Limitations of the ssim quality metric in the context of diagnostic imaging," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 2960–2963.

[322] B. Benzougagh, S. G. Meshram, B. E. Fellah, M. Mastere, M. El Basri, I. Ouchen, D. Sadkaoui, Y. Bammou, N. Moutaoikil, and B. Turyasingura, "Mapping of land degradation using spectral angle mapper approach (sam): the case of inaouene watershed (northeast morocco)," *Modeling Earth Systems and Environment*, vol. 10, no. 1, pp. 221–231, 2024.

[323] S. Chakravarty, B. K. Paikaray, R. Mishra, and S. Dash, "Hyperspectral image classification using spectral angle mapper," in *2021 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*. IEEE, 2021, pp. 87–90.

[324] N. H. Tran, D. Seo, D. Woo, Y. Won, and H. T. Tu, "Alpha cut for interactive image segmentation of thin and elongated objects," *IET Image Processing*, vol. 13, no. 11, pp. 1951–1959, 2019.

[325] R. Giraud and M. Clément, "Superpixel segmentation: A long-lasting ill-posed problem," *arXiv preprint arXiv:2411.06478*, 2024.

[326] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.

[327] N. Keriven, "Not too little, not too much: a theoretical analysis of graph (over) smoothing," *Advances in Neural Information Processing Systems*, vol. 35, pp. 2268–2281, 2022.

[328] S. Tourani, M. H. Khan, C. Rother, and B. Savchynskyy, "Discrete cycle-consistency based unsupervised deep graph matching," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 6, 2024, pp. 5252–5260.

[329] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," in *European Conference on Computer Vision*. Springer, 2020, pp. 681–699.

[330] L. Drumetz, T. R. Meyer, J. Chanussot, A. L. Bertozzi, and C. Jutten, "Hyperspectral image unmixing with endmember bundles and group sparsity inducing mixed norms," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3435–3450, 2019.

[331] X. Sun, L. Yang, B. Zhang, L. Gao, and J. Gao, "An endmember extraction method based on artificial bee colony algorithms for hyperspectral remote sensing images," *Remote Sensing*, vol. 7, no. 12, pp. 16 363–16 383, 2015. [Online]. Available: https://www.mdpi.com/2072-4292/7/12/15834

[332] C.-I. Chang, C.-C. Wu, W. Liu, and Y.-C. Ouyang, "A new growing method for simplex-based endmember extraction algorithm," *IEEE transactions on geoscience and remote sensing*, vol. 44, no. 10, pp. 2804–2819, 2006.

[333] J. M. Nascimento and J. M. Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, 2005.

[334] J. H. Gruninger, A. J. Ratkowski, and M. L. Hoke, "The sequential maximum angle convex cone(smacc) endmember model," in *Proceedings of SPIE*, vol. 5425, no. 1, 2004, p. 14.

[335] T.-H. Chan, W.-K. Ma, A. Ambikapathi, and C.-Y. Chi, "A simplex volume maximization framework for hyperspectral endmember extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4177–4193, 2011.

[336] J. Li and J. M. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," in *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*, vol. 3. IEEE, 2008, pp. III–250.

[337] T.-H. Chan, C.-Y. Chi, Y.-M. Huang, and W.-K. Ma, "A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4418–4432, 2009.

[338] G. Zhou, S. Xie, Z. Yang, J.-M. Yang, and Z. He, "Minimum-volume-constrained nonnegative matrix factorization: Enhanced ability of learning parts," *IEEE transactions on neural networks*, vol. 22, no. 10, pp. 1626–1637, 2011.

[339] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 5, no. 2, pp. 354–379, 2012.

[340] C.-H. Lin, C.-Y. Chi, Y.-H. Wang, and T.-H. Chan, "A fast hyperplane-based minimum-volume enclosing simplex algorithm for blind hyperspectral unmixing," *IEEE Transactions on Signal Processing*, vol. 64, no. 8, pp. 1946–1961, 2016.

[341] A. Ifarraguerri and C.-I. Chang, "Multispectral and hyperspectral image analysis with convex cones," *IEEE transactions on geoscience and remote sensing*, vol. 37, no. 2, pp. 756–770, 1999.

[342] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 3, pp. 650–663, 2004.

[343] D. Hong, L. Gao, J. Yao, N. Yokoya, J. Chanussot, U. Heiden, and B. Zhang, "Endmember-guided unmixing network (egu-net): A general deep learning framework for self-supervised hyperspectral unmixing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6518–6531, 2022.

[344] L. Gao, Z. Han, D. Hong, B. Zhang, and J. Chanussot, "Cycu-net: Cycle-consistency unmixing network by learning cascaded autoencoders," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[345] L. Ren, Z. Han, L. Gao, T. Zhang, R. Wu, and H. Zhang, "Advances in hyperspectral image unmixing: From algorithmic frameworks to practical applications," *Information Geography*, vol. 2, no. 1, p. 100035, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S3050520825000351

[346] D. C. Heinz *et al.*, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE transactions on geoscience and remote sensing*, vol. 39, no. 3, pp. 529–545, 2001.

[347] T. Sainburg, L. McInnes, and T. Q. Gentner, "Parametric umap embeddings for representation and semisupervised learning," *Neural Computation*, vol. 33, no. 11, pp. 2881–2907, 2021.

[348] B. Rusyn, O. Lutsyk, R. Kosarevych, O. Kapshii, O. Karpin, T. Maksymyuk, and J. Gazda, "Rethinking deep cnn training: A novel approach for quality-aware dataset optimization," *IEEE Access*, vol. 12, pp. 137 427–137 438, 2024.

[349] I. Kotaridis and M. Lazaridou, "Remote sensing image segmentation advances: A meta-analysis," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 309–322, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271621000265

[350] Z. Qin, Y. Bai, and Y. Sun, "Ghashing: Semantic graph hashing for approximate similarity search in graph databases," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 2062–2072.