UiT The Arctic University of Norway

Faculty of Science and Technology, Department of Physics and Technology

Extracting Information from Multimodal Remote Sensing Data for Sea Ice Characterization

Torjus Nilsen

FYS-3941 Master's thesis in applied physics and mathematics – 30 ECTS – June 2020

Abstract

Remote sensing is the discipline that studies acquisition, preparation and analysis of spectral, spatial and temporal properties of objects without direct touch or contact. It is a field of great importance to understanding the climate system and its changes, as well as for conducting operations in the Arctic. A current challenge however is that most sensory equipment can only capture one or fewer of the characteristics needed to accurately describe ground objects through their temporal, spatial, spectral and radiometric resolution characteristics. This in turn motivates the fusing of complimentary modalities for potentially improved accuracy and stability in analysis but it also leads to problems when trying to merge heterogeneous data with different statistical, geometric and physical qualities.

Another concern in the remote sensing of arctic regions is the scarcity of high quality labeled data but simultaneous abundance of unlabeled data as the gathering of labeled data can be both costly and time consuming. It could therefore be of great value to explore routes that can automate this process in ways that target both the situation regarding available data and the difficulties from fusing of heterogeneous multimodal data. To this end Semi-Supervised methods were considered for their ability to leverage smaller amounts of carefully labeled data in combination with more widely available unlabeled data in achieving greater classification performance.

Strengths and limitations of three algorithms for real life applications are assessed through experiments on datasets from arctic and urban areas. The first two algorithms, Deep Semi-Supervised Label Propagation (LP) and MixMatch Holistic SSL (MixMatch), consider simultaneous processing of multimodal remote sensing data with additional extracted Gray Level Co-occurrence Matrix texture features for image classification. LP trains in alternating steps of supervised learning on potentially pseudolabeled data and steps of deciding new labels through node propagation while MixMatch mixes loss terms from several leading algorithms to gain their respective

benefits. Another method, Graph Fusion Merriman Bence Osher (GMBO), explores processing of modalities in parallel by constructing a fused graph from complimentary input modalities and Ginzburg-Landau minimization on an approximated Graph Laplacian. Results imply that inclusion of extracted GLCM features could be beneficial for classification of multimodal remote sensing data, and that GMBO has merits for operational use in the Arctic given that certain data prerequisites are met.

Acknowledgements

I would like to express my sincerest gratitude and appreciation to my advisor Associate Professor Andrea Marinoni. Thank you for providing the ideas that began this thesis. Your passion, guidance and expertise in regards to these subjects has served as a great motivation for me over the course of this work.

Thank you to Eduard Khachatrian, Saloua Chlaily, and the rest of the group at the Earth Observation Lab under the Department of Physics and Technology for your help with resolving theoretical and practical problems in a swift manner as well as for being available for technical advice whenever the need arose.

A thanks to my fellow students whose friendship and camaraderie has brought much joy to me during these years in times of both highs and lows. In particular to Bendik and Christian for long days spent together in academia.

Finally, to my friends and family for your continued love and support that made this thesis possible.

Torjus Nilsen,
Tromsø, June 2021.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

While Sea ice has been a defining feature of the arctic for the last 13-14 million years the arctic ocean is now projected to become seasonally ice free by 2040. Periods of lowered amounts of sea ice due to orbital variations has occurred, the last being in the early Holocene, but the recent decline is without equal for the last several thousand years and can not be explained by natural sources. This is expected to cause great changes to European and North American climate, as well as cascading problems for life in the arctic [1]. Apart from climate concerns many countries also conduct operations in the arctic where sea ice poses a challenge for ship traffic and they are therefore dependent on reliable research and monitoring of ice movement [2]. Ice monitoring has therefore long been an important focus of remote sensing.

Modern monitoring of ice activity via plane and satellite became prominent after the second world war with satellite gradually becoming the most widely used method for the last 30 years [2]. Earlier methods included measuring only visible and infrared channels but these were highly dependent on weather conditions and has seen some decline after the introduction of microwave based methods, which in principle can acquire images day and night time without impediments from cloud or lighting situation.

Remote sensing satellites generally measure reflected radiation over a spectrum of wavelengths. With enough of the wavelength spectrum known it is possible to characterize properties of the reflection source, e.g. discerning between different thicknesses of sea ice. The Sentinel-1 is a Synthetic Aperture Radar that captures two polarizations. SAR has the advantage of operating on wavelengths not heavily affected by clouds or lighting, as well as good temporal resolution. Drawbacks of SAR is how prone it is to

problem with scatternoise, e.g. from wind, and its coarse spatial resolution. To overcome these issues, different sources of information could be used in conjunction with the Sentinel-1 data, such as optical sensors. Specifically, optical bands from Sentinel-2 could be a valid option, as SAR and optical data has been used together for accuracy gain in ML classification tasks, e.g. by [3]. Sentinel-2 provides high spatical resolution over 13 optical bands with high potential applicability in the study of snow/ice thickness[4]. Additionally, texture data can be extracted from the bands in the form of a Grey Level Co-occurence Matrix (GLCM). If containing complimentary information using the three modalities, Sentinel-1,-2, and GLCM together could have a positive impact when training a classifier.

Supervised methods could be considered for classification and/or segmentation of the satellite image data as there exist many powerful models in this field [5][6][7]. These will however generally require very large labeled datasets to perform well. Currently there are vast amounts of satellite data from different sensory equipment available, of which very little is labeled. This is because labeling the data manually is as of now typically a very time consuming process which requires great expertise and is inherently subjective to limits/bias of the human observer. It could therefore be of great use to explore routes to automate this process in such a way that all available information, unlabeled and smaller amounts of carefully labeled data acquired by multiple sensors and platforms into multimodal datasets, is utilized to generate fast and accurate classifications [8]. This however also leads to some challenges. In fact, integrating heterogeneous datasets and characterizing the relationships among diverse records is not a trivial task, as it implies the use of higher order moments in data analysis [9]. Intuitively, there will often also be some modes contributing a lot more to overall accuracy than others. Moreover, the degree of reliability and informativity might change across modalities. On the other hand, the benefit is that exploiting small training sets with multimodal analysis could potentially lead to a level of detail greater than that of single modality [10].

In this work, data analysis strategies aiming at tackling the aforesaid challenges have been considered. In particular, data analysis architectures based on semisupervised learning approaches and applied to multimodal remote sensing data have been investigated. The ultimate goal of these schemes is to improve the ability of an automatic learning system to retrieve details on the region of interest by integrating the properties grasped by the heterogeneous sensors, while addressing information extraction in case of limited training sets. In order to retrieve a thorough review of the actual capacity and limitations of this approach, three algorithms based on different analysis

principles have been explored. Specifically, two methods process the multimodal datasets simultaneously [11][12], aiming to take advantage of the diversity of the datasets to obtain a robust understanding of the phenomena occurring on the Earth surface. On the other hand, another method [13] will explore processing of the different modalities separately, fusing the relevant information at a later stage. In this work, the properties of these strategies are explored, so to provide an exhaustive description of the main advantages and drawbacks of these architectures. The aforementioned schemes have been tested on several multimodal remote sensing datasets acquired on sea ice areas and urban scenarios to obtain a reliable assessment of the actual capacity of each scheme. Experimental results show semisupervised learning could be applied to multimodal remote sensing datasets in order to address the scarcity of training datasets that characterizes the operational use of remote sensing data analysis in real life applications.

Chapter 2

Background

This chapter reports on key aspects of remote sensing for sea ice classification and of semi-supervised learning as this is one of the main targets of this thesis.

Since the majority of the experiments fall under SSL algorithms the general problem and solution to training a classifier on split labeled- and unlabeled subsets (X_l, X_u) is introduced and put into perspective with the other leading ML directions of supervised-, unsupervised- and reinforcement learning. Following subchapters presents each of the algorithms used in the experiments, with an overarching description of each algorithm, a short listing of its main components, and a further elaboration on those components.

Because of how the thesis progressed with a focus on image classification for the first two algorithms and image segmentation for the two last the background section and successive chapters have groupings of two, as this seemed the most natural when drawing comparisons. This is however not to say that some of the methods can't have several use cases as e.g. Label Propagation only needs a general undirected fixed graph structure.

2.1 Elements of Remote sensing for sea ice classification

The general purpose of remote sensing is measuring radiation that is backscattered towards a sensor and interpreting it along with how the radiation has been affected from interaction with atmospheric constituents and hit objects[14]. When measured for different wavelengths there are several points of comparison to make up an objects spectral signature. If enough of an ob-

jects spectral distribution is known this can be used to discerning many properties of the targeted object. Remote sensing systems are broadly divided into active and passive sensors. The source of radiation energy might be natural or artificial in the sense of a sensor emitting illumination. Active sensors emits known energy wavelengths towards a target object on earth and measures the energy reflected back. Passive sensors measure energy originating from outside the sensors, e.g. the object itself or an outside source such as the sun.

In this thesis the purpose is separating ice from background such as soil or ocean, and even further to discriminate between different physical- and chemical properties, as well as thickness of ice. Variations in temperature, emissivity, reflectivity and differences from open ocean are all important indicating factors for classifying sea ice. This makes remote sensing from satellites a popular choice for collecting sea ice data, as many of these features are picked up on by specific wavelengths. There are however a number of problems when interpreting the backscattered radiation for classifying objectives. For one, sensitivity to emissivity and thermometric temperature is higly dependent on the selected brightness temperatures of polarization and/or frequencies [4]. Especially during warmer seasons melting ice and the forming of melt ponds often lead to an underestimation of reported ice consentration. In this case, including data from microwave methods help in correcting the result as they can better account for ponding and ice surface status. Another consideration is noise degradation of data, e.g. from atmospheric constituents such as cloud liquid water[4]. A possibility would be correcting for this by estimating cloud liquid water and ice brightness temperature variability, but these are hard to accurately estimate, instead, including frequencies least sensitive to the noise wavelengths are often included. Thin ice can be measured quite accurately from passive microwave. For thicker ice, thickness can not be measured directly from SAR data alone. Ice salinity and roughness is however a good indicator for ice age which SAR can measure. This also provides further basis for including GLCM as it is a texture measure. Seasonal evolution of snow/ice thickness is often studied from surface albedo. This can be deduced from optical data, which is part of the background for including 13 optical bands in the dataset.

One widely available active microwave imaging datatype for sea ice monitoring used today is the Synthetic Aperture Radar or SAR from the Sentinel-1 mission with two bands of different polarization. SAR gained popularity because it remains operable in all weather, day and night with good temporal coverage. A large con of this method is how prone it is to back scatter noise when moving through flow turbulence, wind or vegetation[15]. Passive sen-

sors of Sentinel-2 with its 13 optical bands can then be used to support the data from Sentinel-1. This has been done with accuracy gain on machine learning tasks such as classification in [3]. It is possible to extract further info from the satellite images by calculating the Gray Level Co-occurence Matrix GLCM for each band. This is a texture measure which can in some cases easier pick up on spatial patterns than the bands themselves, and have been used previously for accuracy gain in image classification tasks [16]. If the different modalities (SAR, optical and GLCM) contain complementary data it would intuitively be of interest to leverage as much information as possible for classification in a deep learning setting.

2.2 SSL in the landscape of ML

The traditional engineering approach for designing an algorithmic solution to a problem would often consist of acquiring domain knowledge and using this to create a mathematical model for the physics of the experimental set-up and from this an optimized algorithm that can produce the desired output from an input with some performance guarantees given that the physics model is accurate [17]. The decisions of the optimized algorithm however needs to be specifically programmed and the algorithm will not be focused on a progressively improving but rather just churning out outputs given input.

Machine Learning was revolutionary as instead of acquiring domain knowledge it is focused on the possibly easier task of gathering (or simulating as often is the situation in RL) enough wanted behaviour into a training set and using this to train a computer program how to make its own predictions/decisions when faced with new unseen input based on experience learned through some performance measure feedback from working with the training set [17]. ML is generally thought to consist of four main branches: Supervised Learning, Unsupervised Learning, Reinforcement Learning and Semi-Supervised Learning. Which branch is best when faced with an ML task will depend based on amount of available data, importance of training/inference time, physicality, structure and capture mechanism of the data as well as the overall learning needs for that task. For reasons described below Semi-Supervised Learning was the chosen approach for most of the experiments but it can be good to place SSL into the broader context of the other leading branches in ML.

In a **Supervised Learning** setting all samples have a known groundtruth label and the goal is to learn a mapping function between the input and output spaces. The difference between the expected and actual mapping for

a given point can quantified through a cost function which can be minimized through e.g. gradient descent. More specifically for N pairs of feature vectors X and labels Y that forms the training set s.t.

$$\{(x_1, y_1), \ldots, (x_N, y_N)\}$$

and we wish to learn the function $f : X \rightarrow Y$ that makes $f(x)$ a good prediction for y. This branch has been particularly popular in computer vision due to its efficiency when coupled with convolutional nets but requires large amounts of expertly labeled data.

Unsupervised Learning on the other hand is used when no groundtruth is available and thus seeks to identify groupings inherent to the data [18]. One is therefore expecting there to be hidden interesting patterns in the data that can be exploited in training a model, e.g. for clustering tasks.

Reinforcement Learning has a framework close to the framework in supervised learning with an input frame that is run through a neural network model to produce an output action. The main difference is that RL does not use predetermined structure of a dataset (labeled or unlabeled) to train, but rather starts out with a completely random network that is fed an input frame from an environment and outputs a random action that is sent back to the environment, where the network only receives feedback from the environment after an action is taken. The environment then produces a new input frame based on the past action which together with a reward/penalty based on the current system state is fed back into the agent and this continues in a training loop.

RL has the advantage of letting the agent explore the environment somewhat freely with trial and error through random actions, which can in turn lead to policies with potentially better rewards and behaviors than would be possible in the traditional supervised or unsupervised sense where the model has more of a ceiling in that it can only be trained to be as good as the provided data. For this reason RL has recently seen more popularity in fields like medical delivery systems, robotics and game AIs, as well as natural language processing [19]. RL is most widely used for problems of sequential decision making and/or where it is preferential to simulate an environment rather than gathering real life data and was therefore not considered for the experiments of this thesis.

Semi-Supervised learning (SSL) shares aspects of both supervised and unsupervised learning. Often some labeled data is available but not enough to reliably train a supervised classifier and the majority of data is still unlabeled

as gathering can be both an expensive and time consuming task. If also assuming the distribution of the more abundant unlabeled data has some inherent structure that makes it possible to distinguish samples based on class and this information is complimentary to the labeled data then including this could lead to an increase in classifier performance. This was the direction used in most of the experiments as the overall goal and problem situation of having access to rich amounts of unlabeled data with fewer labeled samples seemed particularly suited towards Remote Sensing problems.

Common assumptions [20] about the underlying data distributions are

- **Manifold Assumption** Data of higher dimensional input space lie on lower dimensional substructures called manifolds that are topological spaces locally resembling of Euclidean space. All sample points belong to a manifold and points lying on the same manifold belong to the same class. By determining all manifolds and which points belong to which manifold it is therefore possible to infer the labels of unlabeled data from the labeled samples.
- **Cluster assumption** Similar points (based on a chosen similarity concept) are more likely to belong to the same class. For objects $X \subset \mathcal{X}$ drawn from input space $\mathcal{X}$ with distribution $p(x)$ a cluster is the set of datapoints $C \subseteq X$ more similar to eachother than other points in X and determining clusters is done by finding a function that maps each input $x \in X$ to a cluster with label $y = f(x)$. Since the distribution $p(x)$ is not known this needs to be approximated from the drawing and chosen concept of similarity.

For SSL image classification the goal is to train a classifier on a data set $X = (x_i)_{i\in[n]}$ of two subsets, $X_l \coloneqq (x_1, \ldots, x_l)$ and $X_u \coloneqq (x_{l+1}, \ldots, x_{l+u})$, where the first subset has labels $Y_l \coloneqq (y_1, \ldots, y_l)$ and the second subset is unlabeled[8]. A general overview of Semi-Supervised Learning can be found in [8].

Chapter 3

Methods

The first two algorithms described are LP and MixMatch and while both fall under SSL they belong to different branches of SSL; label propagation is a transductive graph based approach; MixMatch is based on combining loss terms. The last algorithm, Graph Fusion MBO, is a spectral Semi-supervised method that does not train a model with spectral filters but rather performs classification directly on the data.

3.1 Label Propagation for Deep Semi-supervised Learning

Label propagation in itself is not a newer algorithm and has been used in ML for node labeling by propagating similar nodes through graphs since 2007 [11]. Although the method itself is not considered state-of-the-art because of newer more powerful methods, the recent transductive learning approach where a nearest neighbour graph is constructed from feature embeddings and a model is trained in alternating steps of supervised (on labeled and pseudolabeled samples) and deciding labels of nodes through label propagation makes this method close to or comparable to newer algorithms. Label propagation belongs to a group of pseudo labeling algorithms i.e. SSL methods that gives unlabeled samples a pseudolabel and includes this in training with a supervised loss. Other promising methods in this field include [21][22][23].

The Label propagation algorithm has two main steps. First, a model is trained only on labeled data with supervised loss $L_s(X_L, Y_L)$. A nearest neighbour graph is then constructed from feature embeddings θ of the labeled nodes and all data is included as nodes in the graph. Labeled nodes

are then propagated through unlabeled nodes to generate pseudolabels until all unlabeled nodes on the dataset has a predicted pseudo-label. The model is again trained supervised, but now with labeled and pseudo-labeled loss and dataset s.t. $L_w\left(X, Y_L, \hat{Y}_U\right)$ and this repeats iteratively. Issues of different certainty of predictions and class imbalance are solved by introducing certainty weight w_i and class weight ξ_i for sample x_i. By fusing supervised training, nearest neighbour graph, label propagation and weights iteratively a classifier can be effectively trained on unlabeled data, with particular advantage compared to other SSL classifiers when running on sets with very few sampled data available[24]. Below the preliminaries to and explicit descriptions of the components are stated.

3.1.1 Preliminaries

The trained classifier should map new samples $\mathbb{X}$ to existing class labels by a vector of class confidence scores where each class is predicted with some probability, s.t. $f_\theta \colon \mathbb{X} \to \mathbb{R}^c$ for network parameters θ. This is done by extracting a feature vector $\phi_\theta \colon \mathbb{X} \to \mathbb{R}^d$ from the input and sending the feature vector through fully connected- and softmax layers to get confidence scores. A prediction is made for the highest probable class

$$\hat{y}_i \coloneqq \arg\max_j f_\theta\left(x_i\right)_j \tag{3.1}$$

and j corresponds to the dimension of one of the original classes.

Traditional supervised learning models are trained by minimizing supervised loss, e.g. Cross-entropy $l_s\left(s, y\right) \coloneqq -\log s_y,\ s \in U \quad y \in C$, for a labeled dataset X_L.

$$L_S\left(X_L, Y_L; \theta\right) \coloneqq \Sigma_{i=1}^{l} l_s\left(f_\theta\left(x_i\right) \hat{y}_i\right) \tag{3.2}$$

This term is often included in the total loss for semi supervised models.

If however the dataset has an unlabeled subset with pseudo predicitons $X_U, \hat{Y}_U$, then an additional loss term must be included

$$L_p\left(X_U, \hat{Y}_U; \theta\right) \coloneqq \Sigma_{i=1+l}^{n} l_s\left(f_\theta\left(x_i\right) \hat{y}_i\right) \tag{3.3}$$

Label propagation(transductive diffusion) is about computing a matrix of class predictions Z. Most algorithms today don't do this directly but via an approximation [25] but the what and why to the original problem should still be mentioned. For the extracted feature vectors used earlier $V = \left(v_1, \ldots, v_l,\right.$

$v_{l+1}, \dots v_n\big)$ a symmetric adjacency matrix W can be made to represent how the feature vectors are connected. $W \in \mathbb{R}^{n \times n}$ has positive elements ij between 0 and 1 to show how strongly features v_i and v_j are connected. It is 0 along its diagonal, since a feature vector cannot be connected to itself. To symmetrically normalize the adjacency matrix multiply by the degree matrix $D \coloneqq diag\left(W\mathbf{1}_n\right)$ s.t. $\mathbb{W} = D^{-1/2}WD^{-1/2}$. The final part needed to calculate Z is the label matrix Y of size $n \times c$ and rows of one hot encoded labels where n corresponds to a labeled sample and rows of 0 otherwise. With parameter $\alpha \in [0, 1)$ diffusion matrix Z can be computed by

$$Z \coloneqq \left(I - \alpha\mathbb{W}\right)^{-1} Y \tag{3.4}$$

Predicted pseudo-labels can then be found by choosing the highest probable class j for each row in Z

$$\hat{y}_i \coloneqq \underset{j}{\operatorname{argmax}}\ z_{ij} \tag{3.5}$$

3.1.2 Method

A nearest neighbour graph from a network with parameters θ is described through the set of vertices $V = (v_1, \dots, v_l, v_{l+1}, \dots, v_n)$ and each vertex by $\mathbf{v}_i \coloneqq \phi_\theta\left(x_i\right)$. Connections between vertices are represented in the sparse affinity matrix $A \in \mathbb{R}^{n \times n}$ having elements

$$a_{ij} \coloneqq \left[\mathbf{v}_i^T \mathbf{v}_j\right]_+^\gamma, \text{if } i \neq j \wedge \mathbf{v}_i \in NN_k\left(\mathbf{v}_j\right) 0, \text{ otherwise}$$

and NN_k is the set of k nearest neighbors in X, γ is a parameter from manifold search. Finding affinity matrix of the nearest neighbour matrix for large n is feasible, the full affinity matrix is not. Therefore using $W \coloneqq A + A^T$ is preferred, having symmetric nonnegative properties and zero diagonal.

As mentioned label propagation on the Z matrix directly by equation 3.4 is not the preferred method for large n since the inverse matrix $\left(I - \alpha W\right)^{-1}$ is not sparse and instead conjugate gradient is used to solve the linear system.

$$\left(I - \alpha W\right) Z = Y \tag{3.6}$$

This can be done because $\left(I - \alpha W\right)$ is positive-definite. Pseudo-labels are inferred as $\hat{Y}_U = (\hat{y}_{l+1}, \dots, \hat{y}_n)$ and pseudo-labels are inferred same way as previously.

Different pseudo-label predictions are predicted with differing certainty and pseudo label classes might not be predicted with the same frequency. The chosen loss function should reflect this, which can be done by introducing

certainty and class weights. Certainty weight ω_i of sample x_i consists of a row normalized version of Z s.t. $\hat{z}_{ij} = \frac{z_{ij}}{\sigma_k z_{ik}}$ and maximum possible entropy $\log(c)$. They have the form

$$\omega_i \coloneqq 1 - \frac{H(\hat{\mathbf{z}})}{\log c}$$

Class weight ζ_j are given to class j based on an inverse of the unlabeled-(U_j) and labeled (L_j) populations of that class, written as $\zeta_j \coloneqq (|L_j| + |U_j|)^{-1}$ The total loss for labeled and pseudo-labeled samples becomes

$$L_\omega\left(X, Y_L, \hat{Y}_U; \theta\right) \coloneqq \Sigma_{i=1}^{l}\zeta_u, l_s\left(f_\theta\left(x_i\right), y_i\right) + \Sigma_{i=l+1}^{n}\omega_i\zeta_{\hat{y}_i} l_s\left(f_\theta\left(x_i\right), \hat{y}_i\right) \quad (3.7)$$

For a randomly initialized network with network parameters θ, T training epochs are run on the fully supervised loss in equation 3.2. Feature vectors are extracted, used to generate normalized affinity matrix W and pseudo labels are propagated by 3.5. One epoch is run on the entire training set X with combined pseudo-loss from equation 3.7 and feature extraction/pseudo-loss steps are repeated iteratively.

3.2 MixMatch Holistic SSL

MixMatch Holistic Semi-Supervised Learning [12] is state of the art and mixes the loss terms from many recent SSL approaches: entropy minimization, consistency regularization and traditional regularization. These have the respective benefits of confident predictions on unlabeled data, same output distributions for perturbed inputs and less overfitted models. Although all loss components are important, when discussed in the broader term of SSL methods this could fall under consistency regularization models. MixMatch builds on the works of MixUp [26] with its innovative approach to less confident between boundary predictions for better generalization and shares similarities with other methods such as FixMatch[27] and ReMixMatch[28].

MixMatch in short takes as input two equal sised batches of labeled and unlabeled data, X and U, and outputs augmented versions X' and U'. From augmented labeled and unlabeled batches a loss function is computed by combining three commonly used state-of-the-art loss terms. These are Entropy Minimization, Consistency Regularization and Generic Regularization. The goal in MixMatch is to gain benefits from all three loss terms.

3.2.1 Loss terms

Consistency regularization In Supervised learning data augmentation is often used to increase training data by augmenting already existing labeled data into new ones e.g. by adding noise or translation by a few pixels or other input transformations without changing class semantics. For unlabeled samples in a semi-supervised setting the same intuition should hold s.t. the unlabeled sample x and its augmentation $Augment\,(x)$ should still share the same label, but this needs to be enforced by adding a penalty to the loss term and is called Consistency regularization. In a domain where labeled data is scarce to begin with consistency regularization has been established as a highly important component in other semi-supervised models [29][30][31]. The added loss term in MixMatch for consistency regularization is derived from prior work [32] and is stated as

$$||p_{\text{model}}\left(y|\ \text{Augment}\left(x\right);\theta\right) - p_{\text{model}}\left(y|\ \text{Augment}\left(x\right);\theta\right)||_2^2 \tag{3.8}$$

where the two augment terms differ due to being stochastic transformations.

Entropy minimization Entropy Minimization ensures classes are from more distinct distributions by making sure decision boundaries does not go through high density probability distributions of class subsets. An interpretation of this would be that having more distinct class populations means having more confident predicitions as a result of classes bleeding less into decision boundaries of other classes. This can be done by forcing the classifier to give predictions for unlabeled samples lower entropy. In MixMatch this is achieved by a sharpening function where $\text{Sharpen}\,(p, T)$ for categorical distribution p and a chosen hyperparameter temperature T is defined as

$$\text{Sharpen}\,(p, T)_i := p_i^{\frac{1}{T}} / \Sigma_{j=1}^{L} p_j^{\frac{1}{T}} \tag{3.9}$$

When average class prediction over augmentations found in equation 3.20 is inserted as the categorical distribution and temperature is lowered this leads to lower-entropy predicitons. This comes as a result of 3.9 approaching a Dirac distribution for lower T values.

Traditional regularization is a broad term used for tweaks to a deep neural net that makes it better at generalizing to new unseen data, e.g. dropout and batchnorm, by mitigating its memorization of training data [33]. The authors of the original MixMatch paper [12] applied a weight decay penalizing the L_2 norm of model parameters, as well as MixUp [26]. When training one way of minimizing the empirical risk is to memorize training data and overfit towards the empirical distribution of the training samples. This

in turn leads to bad generalization and a worse performing classifier when applied to samples outside of the training data. MixUp seeks to alleviate this by creating artificial points interpolated from pairs of real datapoints by adding and mixing them by a mixing factor λ s.t. predictions from points between datapoints become less sure. The mixing factor λ comes from a β distribution $\beta(\alpha, \alpha)$ and smoothly approaches the traditional empirical risk minimization as α approaches 0. Prior to MixUp data augmentation is applied to both labeled and unlabeled data. Each labeled sample x_b has one augmented version $\hat{x}_b = \text{Augment}(x_b)$ while each unlabeled sample has K different augmentations applied like so $\hat{u}_{b,k} = \text{Augment}(u_{b,k}), k \in (1, \ldots, K)$.

MixUp trains on convex combination pairs of samples and labels and keeps input and output close. This results in strict linearity among samples. For two samples with labels, (x_1, p_1) and (x_2, p_2), the new artificially interpolated data (x', p') can be generated by

$$\lambda \sim \text{Beta}(\alpha, \alpha,) \tag{3.10}$$
$$\lambda' = \max(\lambda, 1 - \lambda) \tag{3.11}$$
$$x' = \lambda' x_1 + (1 - \lambda') x_2 \tag{3.12}$$
$$p' = \lambda' p_1 + (1 - \lambda') p_2 \tag{3.13}$$

where this differs from the vanilla method on supervised data by adding 3.11 to keep x' closer to x_1 and thus preserving the ordering of the batch after concatenating respectively labeled and unlabeled into the same batch. The combined batch $\mathcal{W}$ is created by shuffling together the two batches of labeled samples with true labels and unlabeled samples with guessed labels

$$\hat{\mathcal{X}} = ((\hat{x}_b, p_b) \;\; b \in (1, \ldots B)) \tag{3.14}$$
$$\hat{\mathcal{U}} = ((\hat{u}_{b,k}, q_b) \;\; b \in (1, \ldots B), k \in (1, \ldots K)) \tag{3.15}$$

and for all sample label pairs in $\hat{\mathcal{X}}$ computing $\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i)$ and adding it to $\mathcal{X}'$ and likewise for unlabeled samples with guesses $\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|})$ and adding it to $\mathcal{U}'$ but now using the rest of $\mathcal{W}$ not used in $\mathcal{X}'$.

3.2.2 MixMatch

Mixmatch uses two inputs, batch $\mathcal{X}$ of labeled one-hot targeted data from L classes, and batch $\mathcal{U}$ with the same size as $\mathcal{X}$ of unlabeled data. This gives two outputs of augmented data, $\mathcal{X}'$ and $\mathcal{U}'$. $\mathcal{X}'$ still shares the same labels as its original batch while $\mathcal{U}'$ has a set of guessed labels. The augmented batches

can then be used to calculate labeled and unlabeled loss as following:

$$\mathcal{X}', \mathcal{U}' = \text{MixMatch}\left(\mathcal{X}, \mathcal{U}, T, K, \alpha\right) \tag{3.16}$$

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \text{H}\left(p, \text{p}_{\text{model}}\left(y|x;\theta\right)\right) \tag{3.17}$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} ||q - \text{p}_{\text{model}}\left(y|u;\theta\right)||_2^2 \tag{3.18}$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}} \tag{3.19}$$

with cross-entropy H between distributions p and q and hyperparameters T, $K, \alpha, \lambda_{\mathcal{U}}$ as previously described.

Labels for the unlabeled data $\mathcal{U}$ are guessed from model predictions by averaging over class distributions for K augmentations of unlabeled samples u_b s.t.

$$\bar{q}_b = \frac{1}{K} \Sigma_{k=1}^{K} \text{p}\left(y|\hat{u}_{b,k};\theta\right) \tag{3.20}$$

and the average is used as the probability distribution input to the previously mentioned sharpening function to generate guessed labels.

The coding implementation for Label propagation and MixMatch continues the pytorch frameworks of [34] [35], extending them to a multimodal dataset of large scale satellite images. When comparing models each run will be affected by choice of hyperparameters, number of labeled data and the supervised model running beneath the SSL algorithm. Oliver et al [29] proposes a set of guidelines for determining these.

For the supervised architecture beneath many SSL algorithms, having the same architecture and not one overly specialized towards a specific algorithm could make comparison between SSL algorithms easier. It should however still be a powerful, widely used, and a reasonable architecture for the type of ML problem. For image classification both [29] and [12] adopts a WideResNet[36] which was considered for both LP and MixMatch. The WideResNet was kept for the MixMatch implementation but was disregarded for the LP implementation as the WideResNet appeared to easily stagnate or heavily underperform during training. The WideResNet used was WRN-28-2 with depth 2 and width 28. The downside to this approach is that the algorithm of MixMatch might not obtain state-of-the-art results, or at least not their highest potential compared to LP, as the latter runs on a net specialized for the image classification task. An 8-layered ConvNet was used for LP after testing of a 16-layered ConvNet lead to overfitting at the cost of testing accuracy. In further work with LP, a deeper network can be reconsidered if larger multimodal datasets obtainable.

[29] found that when giving equal budget to tuning hyperparameters performance gap between SSL and supervised methods is mostly smaller than typically reported. To give algorithms fair comparison several runs on the same amount of labeled data was performed with differing hyperparameters to approach an algorithms best performance case. Ideally the tuning would be further studied, e.g. through black-box hyperparameter optimization or just more runs but this did not fit under the scope of this project. Each run of an algorithm takes considerable time and preferably more runs would be dedicated towards studying hyperparameters for more label data sizes than the ones considered. Some algorithms are more sensitive to labeled data size than others.

The amount of labeled data that has been considered in this work was 120, 180 and 240. In the research papers Deep SSL Label Propagation reported high performance for smaller labeled datasets compared to its peers[11], while MixMatch is generally considered a more powerful approach. It is therefore expected that MixMatch will have higher performance gain compared to Label prop as labeled data is increased.

3.3 The Graph Laplacian

As the Graph Laplacian is a core operator in spectral graph theory and an integral part of the coming spectral based method, Graph Fusion MBO, it is included as a separate subchapter here.

In the continuous domain there is only one definition, i.e. the Laplace-Beltrami operator but in graph domain there are several possible definitions. For a graph with weight matrix W and degree matrix D the normalized symmetric graph Laplacian L is defined as

$$L_s = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

Besides the symmetric normalized Laplacian some other popular ones includes the unnormalized Laplacian and the random walk Laplacian but these are not the focus of this work. Important properties of the graph Laplacian include[37]:

- L_s is symmetric and positive definite.
- The smallest eigenvalue of L is 0 with corresponding eigenvector $1_{|L|}$.
- The smallest eigenvalues multiplicity is the same as the number of connected components in the graph.

The first two are important as they contribute to numerical stability, e.g. since the diffusion step of MBO will always be stable for $\lambda \geq 0$ and SPD guarantees positive eigenvalues. Also for SPD the singular value decomposition coincides with the eigendecomposition which for most programming languages is more numerically stable. Furthermore definite matrices in optimization tasks guarantees the existence of global maxima and minima.

Considering a general function, if the function is smooth then the Laplacian applied to it will also be smooth, and conversely if the function oscillates a lot/has high frequencies then the Laplacian picks up on this and will have high values [38][39]. The eigenvalues and -vectors of the Laplacian, often called Fourier modes, can therefore be seen as more interesting than their Euclidean counterparts as they contain a lot of information related to graph geometry and communities. As a result the eigendecomposition of the Laplacian is very useful for classification and segmentation tasks, as will be seen in later sections on Graph MBO. In MBO a smaller number of eigenvalue/-vector pairs from the Laplacian can be used to approximate the nodespace of the graph from a smaller number of nodes and this serves as input to MBO for node classification.

3.4 Graph based data fusion and segmentation for multimodal images

The Graph Fusion MBO algorithm [13] stands among many other MBO graph node classification schemes based around energy minimization of the Ginzburg-Landau functional after approximating graph Laplacian eigenvectors [40][41][42] often with minor variations of energy function and inclusion of fidelity data. It is part of a newer paradigm in MBO based around a coordinate change[43] that makes comptutation of the diffusion step highly efficient. Some competing options to MBO algorithms for classification of multimode data are graph cuts [44] and other spectral methods such as graph induced learning on subspaces [45] that seeks to improve classification performance by including small amounts of high quality information rich data and aligning it with more abundant low information in a latent shared subspace. The subspace aligned methods especially could make for good comparisons to MBO in future work.

The graph based approach for data fusion and segmentation of multimodal images introduced by Iyer et al[13] is an interesting new approach to classification of multimodal data. It consists of a novel method for fusing of

graph elements from different modes, constructing a weighted adjacency matrix, approximating the spectral decomposition of the graph Laplacian to the adjacency matrix by Nyström, and finally running classification on the decomposition through an iterative MBO scheme alternating between a diffusion and thresholding step which minimizes the Ginzburg-Landau functional. Their MBO implementation is made distinct because of the addition of a semi-supervised term in the energy function that imposes human preference to classes and can generate good classifications from a relatively small amount of known labels, as well as a coordinate change which makes the diffusion step highly efficient. A schematic of the process from multiple input modes to a final classification matrix C is shown in figure 3.1. Components of the semi-supervised graph based data fusion method are explained in detail below.

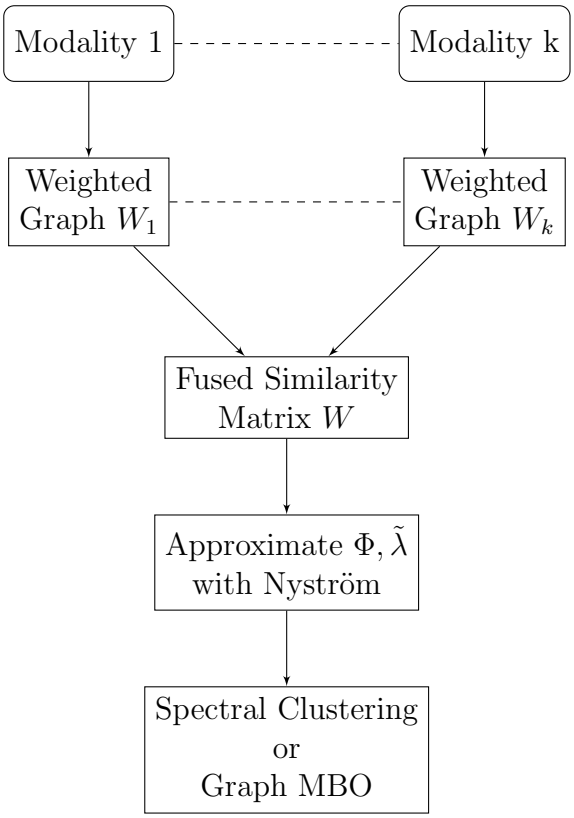


Figure 3.1: Figure showing the work flow of the Graph based data fusion and segmentation model. It takes as input k different modalities captured from sensors of differing physical and geometric properties. The distances for each modalitity is calculated (different distance measures can be chosen for different modes, i.e. gaussian, vector angle measure) and distances are scaled based on their respective modality and used to generate weighted graph representations. Graph representation modes are then fused into one similarity matrix W representing the full input. Landmark nodes are drawn from the fused matrix and used to approximate eigenvalues $\tilde{\lambda}$ and eigenvectors Φ of the graph Laplacian L_{sym} of W to best represent the node space of W. Spectral clustering can be performed directly on the eigenvales/-vectors or they can be used as input to Graph MBO together with a small amount of semi-supervised nodes $\hat{u}$. The final output is a matrix C that classifies the nodes V of W into m separate classes.

3.4.1 Graph fusion

A similarity matrix is a square symmetrical matrix where the ith element of the jth column represents the similarity between the ith and jth nodes

of the graph $G = (V, E)$[46]. Many similarity measures can be used but the radial basis function with scaling parameter σ below

$$w_{ij} = \exp\left(-\text{dist}\left(v_i, v_j\right)/\sigma\right)$$

is a popular choice within machine learning and spectral clustering for data in Euclidean domain. Distance metric is chosen based on what is assumed to best represent distances in the space the data the graph nodes were derived from. Euclidean distance is often reasonable for datapoints in Euclidean space but for e.g. hyperspectral data a vector angle measure could provide a more viable representation[37]. The scaling parameter is included because although depicting the same area different co-registered sets may have vastly different scales depending on their capturing sensor. The sets must therefore first be scaled to make distances comparable prior to fusing. A possibility is including the standard deviation of each set in the expression of their respective graph representation[37]. The scaling factor of modality l is then defined

$$\lambda_l = \text{ std dev}\left(\text{dist}_l\left(x_i^l, x_j^l\right)\right) \quad 1 \leq i, j \leq n$$

which is included in the radial basis function to form the graph representations of each modality. Using the co-registration assumption a single fused graph can be formed from these [13][37]. The number of points is equal for all sets $I = |X_1| = \cdots = |X_l|$ and they share a common indexing where x_i^l references point i in X^l. The notion of distance between collections of graph nodes x_i and x_j is

$$\text{dist}\left(x_i, x_j\right) = \max\left(\frac{\text{dist}_1\left(x_i^1, x_j^1\right)}{\lambda_1}, \ldots, \frac{\text{dist}_k\left(x_i^k, x_j^k\right)}{\lambda_k}\right)$$

,i.e. the weighted maximum across all modes, and this will be the distance measure of each element in our radial basis function as below

$$w_{ij} = \exp\left(-\text{dist}\left(x_i, x_j\right)\right)$$

for the full weighted affinity matrix $W = (w_{ij})_{1 \leq i,j \leq I}$. The intuition behind creating the fused graph out of elementwise maximum distances across sets is that of it possibly being that sets most important discriminative feature for segmentation. An illustration of this would be a dual mode set where the first modality is an elevation dataset and the second is regular RGB imaging. For spectral image segmentation separating grey pavement with similar texture to nearby grey rooftiles could be challenging in RGB domain but trivial in

elevation domain, while similar height items of different colors would have the opposite problem. Since distance need to be similar in both domains for two points to be the same class using the max will ideally remove redundant information while emphazising information unique to a set [13].

A property of each distance dist_l being a formal metric is that the dist on X will also be a formal metric where $\text{dist}(.,0)$ is a norm on the concatenated data set $X^1, \dots X^k$[37]. Consequently the distance metric does not need to be the same across modalities, and it is possible to choose 2 or more separate modalities if a modality is considered unsuited for euclidean space. [13] however generally found that standard euclidean space performed best on image segmentation manifolds.

3.4.2 Spectral Clustering

The general appeal of the spectral clustering algorithms as considered in [47][13][48] is the transformation of abstract datapoints x_i into points $y_i \in \mathbb{R}^k$ by use of the graph Laplacian. This is because inherent properties of transformations using the graph Laplacian will have the effect of enhancing clustering properties of the data, making classification in the new representation trivial[48].

For the adjacency matrix W of the similarity graph the most direct way of partitioning the graph into m subsets $A_1, \dots, A_m$ with minimized similarity between subsets is simply

$$cut\left(A_1, \dots A_m\right) \coloneqq \frac{1}{2}\sum_{i=1}^{m} W\left(A_i, \bar{A}_i\right) \tag{3.21}$$

with $\bar{A}$ denoting the complimentary of A and adjacency matrix between subsets (A, B) being

$$W\left(A, B\right) = \sum_{i \in A, j \in B} w_{ij}.$$

In practice however, minimizing based solely on distinct connections will often result in one large subset and $m-1$ smaller, e.g. many subsets of a single point, which is undesirable. If the volume of subset A_i is defined as

$$vol\left(A_i\right) = \sum_{i \in A, j \in A} w_{ij} = W\left(A, A\right)$$

then the function $\sum_{i=1}^{k}\left(1/Vol(A_i)\right)$ is minimized when all $vol(A_i)$ are equally large and using this in the *cut* will lead to a tradeoff between making sets

of reasonable size and minimized between cluster connection. Introducing the volume function to equation 3.21 we can write the normalized graph NCut[47] as

$$NCut\left(A_1, \dots A_m\right) = \frac{1}{2}\sum_{i=1}^{m} \frac{W\left(A_i, \bar{A}_i\right)}{vol\left(A_i\right)} \tag{3.22}$$

which will decide subset sizes based on edge weights.

It is possible to view graph min-cut as solving for an indicator matrix H where row i corresponds to point i of input data and column m corresponds to class m, s.t. H has dimensions $(i \times m)$ and

$$H_{ij} = \begin{cases} 1, & \text{if } x_i \in A_j \\ 0, & \text{otherwise} \end{cases}$$

Since NCut performs hard classification each row of H will have a single 1. Finding the graph NCut is known to be $O(|V|^{m^2})$[13][49] hard and therefore too computationally intensive to perform on many datasets. A relaxation to the problem proven in [48] is to instead write the NCut with trace Tr

$$\text{NCut}\left(A_1, \dots, A_m\right) = \text{Tr}\left(H^T L_{sym} H\right)$$

using orthogonal matrices

$$\text{argmin}_{Y \in \mathbb{R}^{n \times m}} \text{Tr}\left(Y^T L_{sym} Y\right)$$

where $Y^T Y$ is equal to the identity matrix. For symmetrical L_{sym} and orthogonal Y can be minimized by finding Y from the m eigenvectors of the m smallest eigenvalues. This is used to make an embedding of the abstract datapoints x_i into vectors $y_i \in \mathbb{R}^m$ from the i'th row of Y which is a solution to the relaxed problem. The new featurespace is more suited towards clustering and classification algorithm can be used on top of the eigenvectors to generate a final prediction. Furthermore the obtained eigenvectors can also be valuable for objecdetection as shown in the method section and plays a major part in the graph MBO algorithm.

K-Means [50] can be used for two separate purposes in the following work: as an optional preprocessing step for choosing highly representative landmark nodes prior to Nyström and as a final step in spectral clustering for comparison purposes. In the Nyström approximation sampling nodes at random will often be sufficient for a close approximation to the real eigenvectors [13]. For datasets with properties that are hard to accurately capture from a random sample, e.g. too many classes with low occurring frequency, K-Means can

be used as an unsupervised alternative by choosing landmark nodes from centroids found on the initial unprocessed data.

In terms of spectral clustering K-Means can be used as a final classification into m classes. K-Means is of course not state of the art and as such not expected to perform great, but it serves as a quickly implemented source of comparison. It is expected to provide some indication on the validity of classification based on eigenvectors and may also be able to show the contrast in how a naive method can fail to beneficially merge modes of input captured by sensors with differing statistic and geometric properties as compared to Graph Fusion MBO.

3.4.3 Nyström Extension

The Nyström approximation[40][41][51] is a popular matrix completion algorithm for applications where calculating the full matrix is unfeasible. It has previously seen use in image processing, kernel principle analysis and spectral clustering. In Graph MBO and spectral clustering as explored by this thesis, pixel nodes are segmented based on an approximation to the Graph Laplacian L eigenvectors of W. There are alternative methods to approximating the eigenvectors of L, such as only connecting nearby pixels of the image, making L sparse and thus enabling use of an efficient eigensolver like Lanczos as was considered in [52]. This however has the drawback of not maintaining long range connections between nodes, in addition to approximation properties that are harder to interpret which makes Nyström the preferred option.

The symmetric normalized graph Laplacian L_{sym} and weighted graph representation W are related through the equations

$$D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \phi = \xi \phi$$

$$\begin{aligned} L_{sym}\phi &= \left(1 - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}\right) \phi \\ &= (1 - \xi)\, \phi = \tilde{\lambda}\phi \end{aligned}$$

where $\xi, \tilde{\lambda}$ are matrices with eigenvalues along their diagonals and ϕ is the eigenvector matrix. D is the degree matrix of W. The important result of this is that the eigenvalues $\tilde{\lambda}$ of L_{sym} are equal to the eigenvalues $1 - \xi$ of $D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, and eigenvectors ϕ are shared.

If W is an $n \times n$ matrix where n is the number of nodes the computations of its graph Laplacian involving W will have complexity $\mathcal{O}(n^2)$ and as such

will be quite computationally intensive for large n. Instead Nyström solves an approximate eigenvalue equation by quadrature rule, a way of finding K interpolation weights $a_j(y)$ and K interpolation points $X = \{x_j\}$. The eigenvalue equation is defined as

$$\int_\omega w(y,x)\,\phi(x)\,dx = \gamma\phi(y) \tag{3.23}$$

where $w(y,x)$ is the weight function between points y and x, ϕ is the eigenvector and γ is the eigenvalue. The approximation also introduces an error term $E(y)$ s.t. equation 3.23 becomes

$$\sum_{j=1}^{K} a_j(y)\,\phi(x_j) = \int_\omega w(y,x)\,\phi(x)\,dx + E(y) \tag{3.24}$$

with interpolation weight function $a_j(y)$ over all sampled landmark nodes $K = |X|$.

The error depends on how close the sampled nodes X comes to representing the whole nodespace V[13]. It will be defined more strictly later but is mentioned here as it is an important consideration for sampling of landmark nodes. There are several ways of choosing the landmark nodes X. They can be sampled completely at random which [13] found to be adequate for generating good segmentation results on many modern data fusion segmentation datasets such as the Data Fusion Challenge 2013 and 2015. Datasets of larger scale and with a large number of classes will be harder to accurately represent, in which case "hand picking" a balanced number from each class of semisupervised data is possible. This requires a larger portion of known ground-truth which is not always available for many datasets representing real world problems. A third completely unsupervised option would be choosing landmark nodes from the centers of k-means run on the initial data. Random sampling, k-means or handpicking should therefore be considered on a case-by-case basis after exploration of the dataset.

After sampling X by one of the previously mentioned methods V can be divided into two separate sets: landmark nodes X and remaining nodes Y, to which the following applies $V = X \cup Y$ and $X \cap Y = \emptyset$. By setting $\phi_k(x)$ and λ_k as respectively the k'th eigenvector and eigenvalue, the system of equations for solving these becomes

$$\sum_{x_j \in X} w(y_i, x_j)\,\phi_k(x_j) = \lambda_k \phi_k(y_i) \quad \forall y_i \in Y, \forall k \in 1,\ldots,K \tag{3.25}$$

but this cannot be solved directly without knowing the eigenvectors. Instead, K eigenvectors are approximated through submatrices of W. These are not guaranteed to be orthogonal and must therefore be orthogonalized at a later stage. A possible drawback to this is that the number of calculated eigenvectors cannot be larger than the sampled amount K.

Define W_{XY} as

$$W_{XY} = \begin{bmatrix} w(x_1, y_1) & \cdots & w(x_1, y_{N-L}) \\ \vdots & \ddots & \vdots \\ w(x_L, y_1) & \cdots & w(x_L, y_{N-L}) \end{bmatrix} \tag{3.26}$$

i.e. connections from nodes in X to nodes in Y. Likewise W_{XX} and W_{YY} are the connections from nodes of the first subscript to nodes of the second subscript, and $W_{YX} = W_{XY}^T$ if W is undirected. The weighted affinity matrix $W \in \mathbb{R}^K \times \mathbb{R}^K$ and its eigenvectors $\phi \in \mathbb{R}^K$ can then be rewritten by submatrices into

$$W = \begin{bmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YY} \end{bmatrix}$$

and $\phi = \begin{bmatrix} \phi_X^T \phi_Y^T \end{bmatrix}$. Now we wish to express equation 3.25 in matrix form through these submatrices and must therefore first apply spectral decomposition to W_{XX}

$$W_{XX} = B_X \Gamma B_X^T$$

where B_X is an $X \times X$ matrix of eigenvector columns and Γ is the diagonal matrix of eigenvalues. Using this new notation for equation 3.25 we get

$$B_Y = W_{YX} B_X \Gamma^{-1}$$

and the full approximation of eigenvectors in W becomes $B = [B_X B_Y] = [B_X W_{YX} B_X \Gamma^{-1}]$ with related approximation of W

$$W = \begin{bmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YX} W_{XX}^{-1} W_{XY} \end{bmatrix}$$

If $|Y| \gg |X|$ then the majority of computations lie in W_{YY} but as shown above this can be approximated using the much smaller submatrices W_{XX}, W_{XY}, with a known error of $||W_{YY} - W_{YX} W_{XX}^{-1} W_{XY}||$ [40][51]. This is the major benefit of Nyström as the extension only has computational complexity of approximately $\mathcal{O}(n)$[40] but instead of using this directly the approximation should first be orthogonalized.

Consider an arbitrary unitary matrix A and diagonal matrix Ξ s.t.

$$\Phi = \begin{bmatrix} W_{XX} \\ W_{YX} \end{bmatrix} \left(B_X \Gamma^{-1/2} B_X^T \right) \left(A \Gamma^{-1/2} \right)$$

Diagonalize W by making A unitary, i.e. $\Phi^T \Phi = \mathbf{1}$ and rewriting $W = \Phi \Gamma \Phi^T$ to get an expression with correct A

$$\Phi^T \Phi = \left(Y^T \right)^{-1} W_{XX} Y^{-1} + \left(Y^T \right)^{-1} W_{XX}^{-1/2} W_{XY} W_{YX} W_{XX}^{-1/2} Y^{-1} \tag{3.27}$$

where $Y = A\Gamma^{-1/2}$. Multiply the rightside of equation 3.27 by $\Gamma^{1/2} A$ and leftside by its transpose

$$A^T \Gamma A = W_{XX} + W_{XX}^{-1/2} W_{XY} W_{YX} W_{XX}^{-1/2} \tag{3.28}$$

and finally output the columns of Φ and diagonal elements of Γ as respectively the i'th eigenvector and -value pairs.

If L_s is to be used further for segmentation purposes W should also be normalized. This is done by

$$\begin{aligned}
\begin{bmatrix} d_X \\ d_Y \end{bmatrix} &= \begin{bmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YX} W_{XX}^{-1} W_{XY} \end{bmatrix} \begin{bmatrix} \mathbf{1}_K \\ \mathbf{1}_{N-L} \end{bmatrix} \\
&= \begin{bmatrix} W_{XX} \mathbf{1}_K + W_{XY} \mathbf{1}_{N-L} \\ W_{YX} \mathbf{1}_K + \left(W_{YX} W_{XX}^{-1} W_{XY} \right) \mathbf{1}_{N-L} \end{bmatrix} \\
s_X &= \sqrt{d_X} \\
s_Y &= \sqrt{d_Y} \\
\overline{W}_{XX} &= W_{XX} ./ \left(s_X S_X^T \right) \\
\overline{W}_{XY} &= W_{XY} ./ \left(s_X s_Y^T \right)
\end{aligned}$$

where $\mathbf{1}_K$ is the K-dimensional unit vector and $A./B$ denotes the componentwise divison between matrix elements in A and B. Algorithm 1 summarizes the steps in a pseudocode following the general layout in [40] with slight changes to fit better with the following segmentation.

Algorithm 1 Nyström approximation

Input: A set of features $V = \{x_i\}_{i=1}^{X}$

Output: K eigenvalue-eigenvector pairs $\left(\phi_i, \tilde{\lambda}_i\right)$ where ϕ_i is the ith column of Φ and $\tilde{\lambda}_i = 1 - \xi_i$ is the ith diagonal element of Ξ

1: Partition the set Z into $Z = X \cup Y$ (by one of the discussed drawing methods), where X consists of L selected elements.
2: Calculate W_{XX} and W_{XY} by equation 3.26
3: $d_X = W_{XX}\mathbf{1}_L + W_{XY}\mathbf{1}_{N-L}$
4: $d_Y = W_{YX}\mathbf{1}_L + \left(W_{YX}W_{XX}^{-1}\right)\left(W_{XY}\mathbf{1}_{N-L}\right)$.
5: $s_X = \sqrt{d_X}$ and $s_Y = \sqrt{d_Y}$.
6: $W_{XX} = W_{XX}./\left(s_X s_X^T\right)$.
7: $W_{XY} = W_{XY}./\left(s_X s_Y^T\right)$.
8: $B_X \Gamma B_X^T = W_{XX}$ (using the SVD).
9: $S = B_X \Gamma^{-1/2} B_X^T$.
10: $Q = W_{XX} + S\left(W_{XY}W_{YX}\right)S$.
11: $A\Xi A^T = Q$ (using the SVD).
12: $\Phi = \begin{bmatrix} B_X\Gamma^{1/2} \\ W_{YX}B_X\Gamma^{-1/2} \end{bmatrix} B_X^T \left(A\Xi^{-1/2}\right)$ diagonalizes W.

Note that for lines 8 and 11 of algorithm 1 the singular value decomposition and eigendecomposition coincide since W_{XX} and Q are symmetric positive definite matrices.

3.4.4 Semi-supervised MBO Classification

Much recent work has been done on classification of graph nodes by use of the Merriman-Bence-Osher scheme[40][41][42]. The contribution of this thesis mainly considers a particular direction of the multimode semisupervised segmentation MBO based on a coordinate change involving the eigendecomposition of the graph Laplacian that greatly improves efficiency of the diffusion calculation of the Ginzburg-Landau functional in the MBO, as explored previously by [13][43].

As in [13] make u the assignment matrix, similar to the H of NCut, of dimension $\mathbb{R}^i \times \mathbb{R}^m$ where the i-th row represents point i and the m-th row element represents how closely associated that point is to class m. Elements of u for intermediary diffusion half-steps will be real valued probabilites where the largest is chosen during thresholding and when the MBO has converged the thresholding of the final iteration is returned as the classification vector

C_i. u can be initialized either by randomly assigning a 1 to each row or by setting all row elements to $\frac{1}{m}$. The energy function consists of three terms: the Dirichlet energy, the multiwell potential and a fidelity term, and is defined

$$\begin{aligned} E(u) &= \epsilon \cdot \mathrm{Tr}\left(u^T L_{\mathrm{sym}} u\right) \\ &+ \frac{1}{\epsilon}\sum_i W(u_i) \\ &+ \sum_i \frac{\mu}{2}\chi(x_i)\,||u_i - \hat{u}_i||_{L_2}^2 \end{aligned} \tag{3.29}$$

where the multiwell potential with standard basis vector e_k is

$$W(u_i) = \Pi_{k=1}^m \frac{1}{4}||u_i - e_k||_{L_1}^2 \tag{3.30}$$

and $\hat{u}$ represents the fidelity data points of the semi-supervised input s.t.

$$\chi(x_i) = \begin{cases} 1, & \text{if } x_i \text{ is part of the semisupervised input} \\ 0, & \text{otherwise} \end{cases}$$

The first and second terms of equation 3.29 approximates the classical Ginzburg-Landau and as ϵ approaches 0 these will converge to the total graph variation norm[53] as shown below

$$TV(u) = \sum_{i,j} w_{ij}|u_i - u_j| \tag{3.31}$$

Minimizing the energy function by gradient descent is done through the Allen-Cahn equation with an additional semi-supervised term, i.e.

$$\frac{\partial u}{\partial t} = -\epsilon L_{\mathrm{sym}} u - \frac{1}{\epsilon}W'(u) - \frac{\partial F}{\partial u} \tag{3.32}$$

where F is the third term of equation 3.29 s.t.

$$\frac{\partial F}{\partial u} = \frac{\partial \frac{\mu}{2}\chi(x)(u-\hat{u})^2}{\partial u} = \mu\chi(x)(u-\hat{u})$$

which inserted into 3.32 yields

$$\frac{\partial u}{\partial t} = -\epsilon L_{\mathrm{sym}} u - \frac{1}{\epsilon}W'(u) - \mu\chi(x)(u-\hat{u}) \tag{3.33}$$

A variation of solving this with the MBO scheme[13] is to iteratively minimize the energy in two separate steps and letting u^n denote the prediction of

iteration n. The first step minimizes terms 1 and 3 of the energy equation by diffusion and the second step minimizes the second term by thresholding on the probabilities calculated in the diffusion step. Starting from the diffusion equation

$$\frac{u^{n+\frac{1}{2}}-u^n}{dt} = -L_{sym}u^{n+\frac{1}{2}} - \mu\left(u^{n+\frac{1}{2}}-\hat{u}\right) + (1-\chi(x))(u^n-\hat{u}) \tag{3.34}$$

the next half-step is calculated by solving for $u^{n+\frac{1}{2}}$

$$\begin{aligned} u^{n+\frac{1}{2}}-u^n = &-L_{sym}u^{n+\frac{1}{2}}dt - \mu\left(u^{n+\frac{1}{2}}-\hat{u}\right)dt \\ &+ (1-\chi(x))(u^n-\hat{u})\,dt \end{aligned}$$

$$u^{n+\frac{1}{2}} + L_{sym}u^{n+\frac{1}{2}}dt + \mu u^{n+\frac{1}{2}}dt = u^n + \mu\hat{u}dt + (1-\chi(x))(u^n-\hat{u})\,dt$$

and writing out the last term to get

$$u^{n+\frac{1}{2}} + L_{sym}u^{n+\frac{1}{2}}dt + \mu u^{n+\frac{1}{2}}dt = u^n + \mu\hat{u}dt + (u^n-\hat{u})\,dt - \chi(x)(u^n-\hat{u})\,dt \tag{3.35}$$

The eigendecomposition of the symmetric Graph Laplacian L_{sym} is

$$L_{sym} = H\Lambda H^T \tag{3.36}$$

where H is the matrix of columnwise eigenvectors and Λ is the diagonal matrix with eigenvalues along its diagonal. This can be used to write the following coordinate changes

$$\begin{aligned} u^n = Ha^n, \quad \chi(x)(u^n-\hat{u}) = Hd^n, \\ \hat{u} = -\frac{Hd^n}{\chi(x)} + u^n = -\frac{Hd^n}{\chi(x)} + Ha^n \end{aligned}$$

which when applied to the left side of equation 3.35 becomes

$$\begin{aligned} &Ha^{n+\frac{1}{2}} + H\Lambda H^T Ha^{n+\frac{1}{2}}dt + \mu Ha^{n+\frac{1}{2}}dt \\ =&Ha^{n+\frac{1}{2}} + H\Lambda a^{n+\frac{1}{2}}dt + \mu Ha^{n+\frac{1}{2}}dt \\ =&Ha^{n+\frac{1}{2}}(1+\Lambda+\mu dt) \end{aligned}$$

and right side of equation 3.35

$$\begin{aligned} &Ha^n + Ha^n dt + \mu\left(-\frac{Hd^n}{\chi(x)} + Ha^n\right)dt - \left(-\frac{Hd^n}{\chi(x)} + Ha^n\right)dt - Hd^n dt \\ =&Ha^n - \mu\frac{Hd^n}{\chi(x)}dt + \mu Ha^n dt + \frac{Hd^n}{\chi(x)}dt - Hd^n dt. \end{aligned}$$

Notice that all terms of both the left and right side contain H and use this so simplify further by using that $H^T H = I$ and $\chi(x) = 1$ for fidelity points

$$H^T H a^{n+\frac{1}{2}} (1 + \Lambda + \mu dt) = H^T H \left(a^n - \mu d^n dt + \mu a^n dt + d^n dt - d^n dt\right)$$

and finally factoring out and dividing by $(1 + \Lambda + \mu dt)$ we get

$$a^{n+\frac{1}{2}} = \frac{(1 + \mu dt)\, a^n - \mu dt \cdot d^n}{1 + \mu dt + dt\Lambda}$$

Now the diffusion step for eigenvalue k, λ_k, in ascending order can be calculated as

$$a_k^{n+1} = \frac{(1 + \mu dt)\, a_k^n - \mu dt \cdot d_k^n}{1 + \mu dt + dt\lambda_k} \tag{3.37}$$

which will always be stable if $\lambda_k \geq 0$ for $k = 0, \ldots K$, as long as μ and dt are chosen to be positive. The choice of μ is generally found through trial when exploring a specific dataset. It is representative of the expected quality of the semi-supervised input and is often therefore set quite high, e.g., $\mu \geq 10^3$[43]. Heuristically dt can be thought of as akin to the learning rate in neural nets. It is a hyperparameter that controls model runtime and the final accuracy of the model where the two are inversely related. After diffusing u^n will be a matrix of real valued probabilities. The second step thresholds on the probabilties, choosing the class with the highest probability for each row i.e.

$$u_i^{n+1} = e_r \quad \text{where} \quad r = \text{argmax}_j u_{ij}^{n+\frac{1}{2}} \tag{3.38}$$

The diffusion step is performed once between each thresholding in [13], but it can also be repeated s times before thresholding as done in [43].

For larger graphs calculating all the eigenvectors is unfeasible. As the eigenvectors belonging to the smallest eigenvalues will have the least computational significance one can instead approximate a smaller number of eigenvectors and use this as a truncated substitute for H in equation 3.36, making the MBO model much less computationally intensive. A recap of the method is shown in algorithm 2.

Algorithm 2 Semi-Supervised MBO

Input: Graph embedded input u and semi-supervised input $\hat{u}$ both dim $\mathbb{R}^{i\times m}$. Parameters s, μ, dt.

Output: Matrix $C \in \mathbb{R}^{i\times m}$ of class predictions

Approximate $k \ll i$ smallest eigenvalues λ and eigenvectors H of the symmetrical graph Laplacian.
Initialize u^0 randomly and $d_k^0 = 0$ for all k
while Purity$(u^{n+1}, u^n) < 99.99\%$ **do**
    $a^n = H^T u^n$
    for $j \leftarrow 1$ to s **do** ▷ Step1: Diffuse s times
        $a_k^{n+1} = \frac{(1+\mu dt)a_k^n - \mu dt \cdot d_k^n}{1+\mu dt+dt\lambda_k}$ for $k = 1$ to $k = K$
        $u^{n+\frac{1}{2}} = Ha$
        $d^n = H^T\chi(x)\,(u^n - \hat{u})$
    end for
    $u_i^{n+1} = e_r, \quad r = \text{argmax}_j u_{ij}^{n+\frac{1}{2}}.$ for $i = 1$ to $i = I$ ▷ Step 2: threshold
    $n \leftarrow n + 1$
end while
return last u^n as the final classification C

Chapter 4

Experimental Results

This chapter starts by presenting the different datasets used in the experiments. The first two methods, LP for DSSL and MixMatch, were used for semi-supervised classification of satellite images with additional extracted GLCM texture features where the modalities are processed simultaneously and they are therefore compared against each other, weighing potential benefits and drawbacks of each method and also including an ablation study into the relative importance of each methods optional modules. Results on Graph Fusion MBO is shown and evaluated last as it focused on processing of different modalities and fusing at a later stage for image segmentation of multimodal satellite images.

4.1 Dataset North-East Svalbard

The dataset used in LP and MixMatch for image classification tasks and later in Graph Fusion MBO consists of multi-sensor sea ice images of southern Svalbard captured by the Sentintel-1 and Sentinel-2 earth observation missions. The first 13 optical bands are from Sentinel-2 and covers visible, near infrared, and short infrared parts of the spectrum. The last two bands is from two polarization bands of Sentintel-1, HH and HV.

Names of bands are as following:

- Band 1 - Coastal aerosol
- Band 2 - Blue
- Band 3 - Green

- Band 4 - Red
- Band 5 - Vegetation red edge
- Band 6 - Vegetation red edge
- Band 7 - Vegetation red edge
- Band 8 - NIR
- Band 8A - Narrow NIR
- Band 9 - Water vapor
- Band 10 - SWIR-CIRRUS
- Band 11 - SWIR
- Band 12 - SWIR
- HH Polarisation
- HV Polarisation

Below is a short introduction to some common uses for the different bands.

The visible spectrum, $0.38 - 0.72$µm made up by the blue, green and red bands, is a small part of the spectrum width, yet has traditionally been and still is an important part of remote sensing. It is however affected by similar limitations as the human visual system, e.g. measurements in this range can be next to useless due to noise from atmospheric interaction or clouds, and as such is often not enough alone for remote sensing classification tasks.

The NIR(Near Infrared) channels $0.76 - 0.90$µm are often used for plant structure and more importantly for this ice/water classification task for borders between water and other features, as water completely absorbs radiation in this wavelength.

Short wave infrared or SWIR bands have wavelengths of $1000 - 3000$nm and has application in discerning moisture, mineral content and different types of snow/ice.

Additionally, 14 Gray Level Co-occurrence Matrix(GLCM) textural features have been extracted from each band of both sensors. The GLCM matrix is a texture measure of difference between spatial brightness values, giving insight to possible spatial patterns. The motive for including this additional information is that the regular bands can measure chemical properties of the ground, while the GLCM can pick up information on possible spatial patterns that the bands might not provide. The different information is therefore

not necessarily dependent and could be complementary to each other thus resulting in a higher accuracy than just the band images alone [16]. For a general description of GLCM and its uses [16] is also a good source. GLCM has been used previously with accuracy gain for image classification tasks, e.g. by [54][55][56].

The bands have the following GLCM attributes

1. Band/Polarisation
2. Angular second Moment(Energy)
3. Contrast
4. Correlation
5. Variance
6. Inverse difference Moment(Homogeneity)
7. Sum Average
8. Sum Variance
9. Sum Entropy
10. Entropy
11. Difference Variance
12. Difference Entropy
13. Information Measure of Correlation I
14. Information Measure of Correlation II
15. Maximal Correlation Coefficient

The NE Svalbard dataset consists of 1458 by 1830 pixels, with 15 bands times 15 attributes for a total of 225 channels, or $(1458, 1830, 225)$. Each pixel is originally labeled a class between 0 to 5, for a total of 6 classes. The classes are Background, Grey Ice, Grey-White Ice, Open Water, Thick First year ice and Thin first year ice. An image description is shown in figure 4.1 where one pixel on the image corresponds to 60 meters in real life.

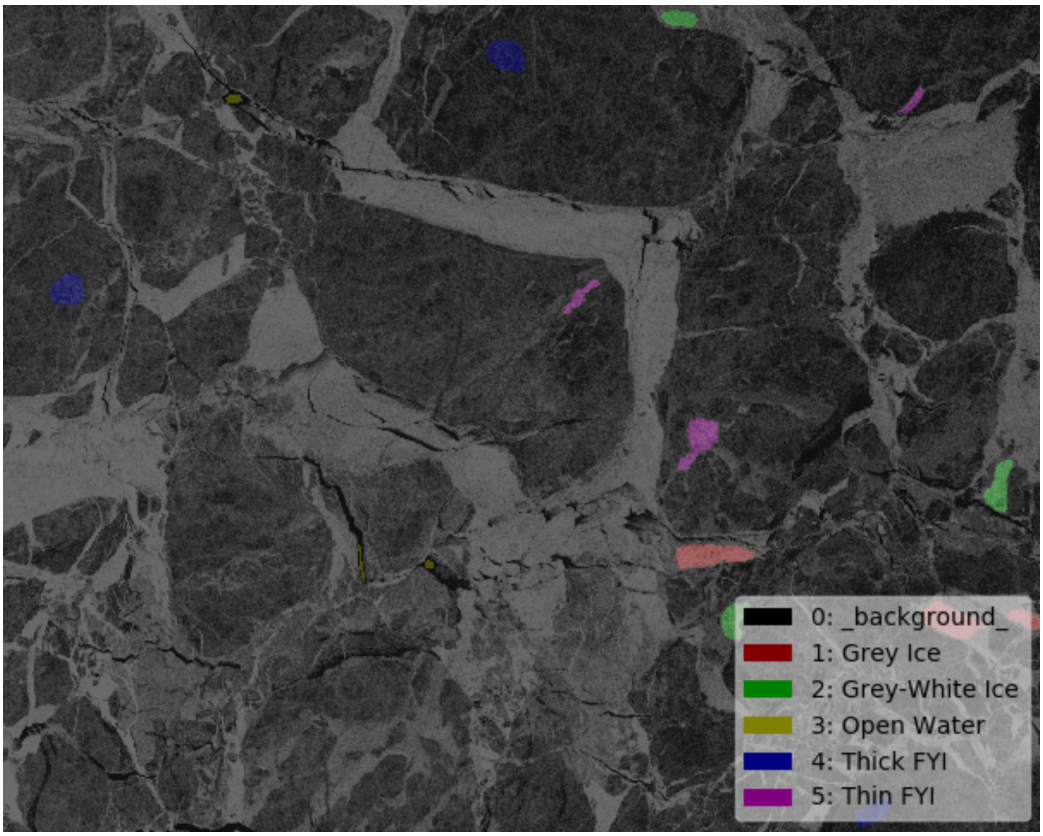


Figure 4.1: Description image of the North-East Svalbard multimodal dataset captured by the Sentinel 1 and Sentinel 2 missions. The image shows 6 classes, of which 1 is background and 5 are different ice types. Background makes up most of the image with scattered clusters of ice.

The first two methods this project aims to explore, Label propagation and MixMatch, are image classification methods while the data is more geared towards an image segmentation problem. Therefore either the models must be adapted to the dataset or the dataset must be adapted to the models. Because there is little to no guarantee that an image classification SSL method will work well on segmentation problems the latter was chosen for these methods. Smaller images of size 32 by 32 was cut out of the original image with an overlap/stride of 4 pixels. These smaller image patches were labeled as the majority ice class (not background), given that enough pixels of the majority label occurred in the image patch were found. Cut off was set to 1/6 of an image being that class as this seemed a fair amount after inspection of samples. An interesting continuation for further work would be comparing this with a pixel-wise segmentation and seeing if the Label propagation

and/or MixMatch are able to obtain similar or better results than image wise classification to an adapted dataset.

Although there are many possible (200.000+) image patches to create a dataset from, the vast majority of these would be labeled as background. The classes should ideally be somewhat balanced so one class does not dominate training. Some amount of background labeled images saved was therefore not included in the dataset since having much greater amount of this class than all other classes might not provide much benefit. After this, remaining difference between classes can be accounted for by using class weights or by duplicating the less occurring classes until having an even distribution, which has been found to be beneficial for training neural nets [57]. There is no easy way of telling exactly how much data a deep learning algorithm needs to perform well, but some insight can be gained by increasing amount of training samples and seeing if train error increases while test error decreases. This would mean that the model is harder to fit to training data and better generalizes to test data as a result. After running models on the original data the training accuracy converged towards 100, a clear sign of overfitting. To mitigate this the dataset was artificially increased through data augmentation.

4.2 Dataset Southern Svalbard

The dataset Southern Svalbard consists of 2 SAR polarizations captured by Sentintel 1 and 2 passive microbands HH and HV from AMSR. It spans 7955 by 10365 pixels by 4 channels/bands and as such is both the largest dataset and the one with the least amount of bands. Figure 4.2 shows a description image of the Southern Svalbard dataset with mainly background and smaller clusters of brash ice, first-year ice and open water spread around, similar to the NE Svalbard set but with significantly more class clusters occurring.

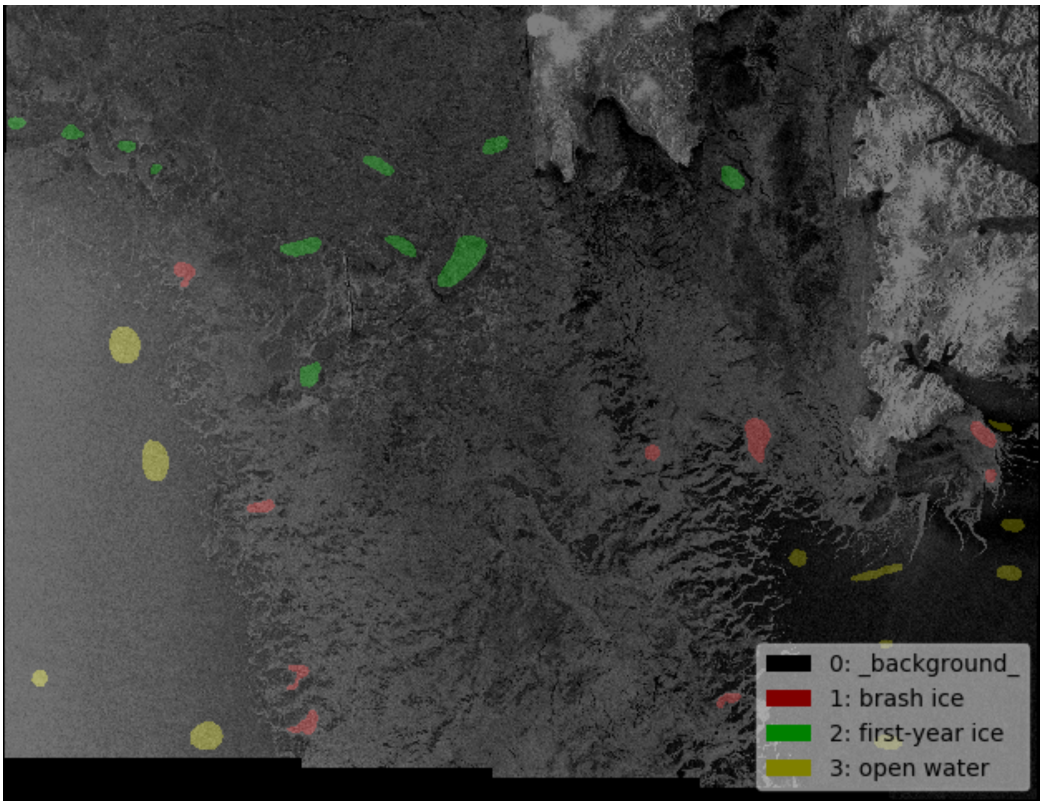


Figure 4.2: Description image of the Southern Svalbard multimodal dataset captured by the Sentinel 1 and AMSR missions. The image shows 4 classes, of which 1 is background and 5 are different ice types. Background makes up most of the image with scattered clusters of ice.

4.3 Dataset Trento

The first dataset used for pixel node classification is the Trento dataset which depicts countryside landscape from outside of Trento, Italy. It consists of 600 by 166 pixels from two types of sensors where the first 7 bands are LIDAR DSM data captured by an Optech ALTM 3100EA sensor and the next 70 bands are hyperspectral data from an AISA Eagle sensor for a total of 77 bands. The hyperspectral bands have wavelengths ranging from 402.89 to 989.0 nm and spectral resolution of 9.2 nm. Both sensors have a spatial resolution of 1 m. There are 7 classes in the dataset, i.e. Background, Apple tree, Building, Ground, Wood, Vineyard and Water. Figure 4.3 shows several interpretations of the Trento dataset. Looking at the testing data in image four of that figure one can see that most classes have large distinct clusters

while the Ground and Water classes consists of thinner stretches.

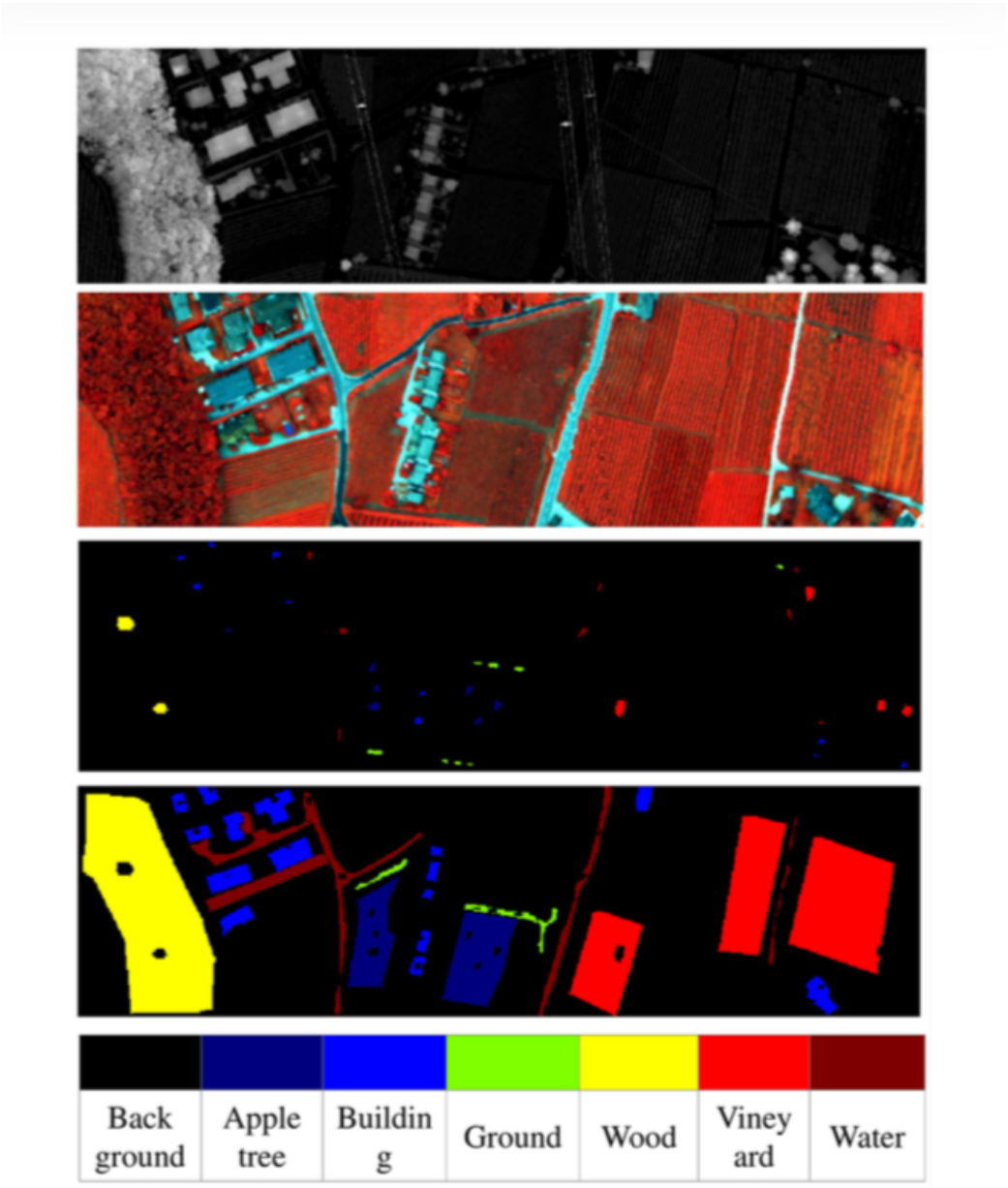


Figure 4.3: A figure showing different aspects of the Trento dataset. It shows respectively a LIDAR raster data image, an RGB visualization of the hyperspectral bands where red is represented by band 40, blue by 20 and green by 10, groundtruth feature maps for training and testing samples, as well as a colordescription of the classes.

4.4 Dataset Houston

The Houston dataset was part of the 2013 IEEE Geo Science and Remote Sensing Data Fusion Contest. It is an urban dataset showing the campus and surroundings of the University of Houston campus. The first seven bands are again LIDAR DSM images while the last 144 bands are hyperspectral imagery ranging from 380 nm to 1050 nm. The spatial resolution of both the LIDAR and hyperspectral modalities are 2.5 m. It has significantly more classes than most other explored sets at 16 classes.

4.5 Ablation study: LP and MixMatch

The separate Sentinel-1, Sentinel-2 and GLCM data was evaluated over 80 epochs for the supervised phase 1 of LP to compare how much they contributed to classification. The mean accuracies were respectively 72.99, 72.83 and 81.83. S1 and S2 had nearly identical mean accuracies, while GLCM had quite a lot higher mean, which gives some basis for including this in the mixed dataset.

Table 4.1 shows the ablation results for Deep Label propagation. Overall the error rate was quite high for this algorithm. Out of the tested components the L2 normalization seemed to have the clearest impact on classificaiton, as the error rate from baseline LP increased by 2.71 and 1.97 for respectively 120 and 240 labels. Adding the Mean teacher exponential moving average seemed to give a slight boost to accuracy, but it does not deviate enough from the regular LP implementation to confirm this. What is not reflected from the error rate table is however that MT EMA resulted in generally higher phase 1 accuracy, as well as more stable learning. Decreasing the labeled batch size makes the model depend more on unlabeled data. This hyperparameter did not seem to make a large difference on results. It is possible that the contributions of both MT and size of labeled samples per batch towards lowered error rate would be more pronounced for a better tuned set of hyperparameters or a larger dataset. A large problem for LP was obtaining a good phase 1 as a starting point for phase 2. During early testing it was found that e.g. too few or too many training epochs in phase 1 could lead to very stagnant learning, or models not learning at all. A possible reason for this is that the learned parameters θ from phase 1 used in creating the nearest neighbour graph must lead to a good general node representation that can distinguish class. If not trained enough it is likely that the learned parameters does not yet give a good class representation, while if trained for too long ($\tilde{1}00$ epochs+) testing accuracy seemed to decline in favor of

training accuracy due to overfitting.

Ablation	120 labels	240 labels
Label propagation	13.54	10.95
Label propagation, without L2 normalization	16.25	12.92
Label propagation, with Mean teacher	11.46	10.63
Label propagation, labeled batchsize = 15	12.51	11.79
Label propagation, labeled batchsize = 10	11.67	12.50

Table 4.1: Table showing error-rates of Label Propagation on the multimodal dataset SE Svalbard. Ablations were done for 120 and 240 labeled samples.

Ablations for MixMatch are shown in table 4.2. The regular MixMatch implementation outperformed the LP one with error rates of 9.39 and 5.40 for respectively 120 and 240 labeled samples. MixMatch appeared more sensitive than LP towards amount of labeled samples, as error rate at 240 labels was half than that of LP. The Temperature sharpening and MixUp components were compared to the barebones approach. Both Sharpening and MixUp contributed towards lower error rate, but MixUp considerably more so. When removing MixUp the error rate of MixMatch was comparable to regular LP.

Ablation	120 labels	240 labels
Mixmatch	9.39	5.40
MixMatch, without temperature sharpening T=1	10.33	7.04
Mixmatch, without MixUp	12.44	8.22

Table 4.2: Table showing error-rates of MixMatch on the multimodal dataset NE Svalbard presented in Section 4.1. Ablations were done for 120 and 240 labeled samples.

4.6 Graph based data fusion and segmentation for multimodal images

4.6.1 Fusing

Implementation-wise the fusing operation was straightforward, as constructing the affinity between weighted nodes from a Euclidean basis function is not as inclined as the Cosine distance measure in [43] to have infinite/negative similarity weights and therefore have well defined weighted graphs for all nodes. As such problems arising from the fusing operations in these experiments were not as easy to identify as the final classification maps are dependent also on the Nyström and MBO modules and possible difficulties due to the fusing module are more subtle. From the results however it can be reasonably expected that a significant part of the problems when classifying datasets with higher numbers of bands/modalities such as the Houston set was due to the fusing operation discarding too much of the data needed to generate accurate classifications. The Graph Fusion is an intuitive method for potentially fusing complimentary data and removing redundancies. When the data is too large to effectively condense down into a fused graph however it may be quite desctructive since all but one mode of information is discarded per node.

Another concern is that a maximum betwen all distances as a similarity measure is not necessarily the best to distinguish class in many real world datasets. It can at times be since samples need to be similar for every modality to be classified as the same class. E.g. if color similarity of a hyperspectral modality and height and shape data of a LIDAR modality both have complimentary data relevant to classification, as worked well in the Trento dataset. This is however not guaranteed by any means to transfer well to other datasets.

In many cases it can also be thought that similarity between modalities can be helpful for classification tasks but not necessarily enough on its own and needs to be coupled with supplementary distinguishing factors for drawing class boundaries. For datasets where ground truth is unknown one may not necessarily know enough to settle on maximum (or other similarity measures) as a good source of discrimination for distinguishing classes. Usage of the fusing operation therefore requires careful consideration and knowledge about each dataset.

4.6.2 Nyström

A challenge with the Nyström approximation was numerical stability when manipulating the fused weighted affinity matrices for landmark nodes W_{XX} and from landmark nodes to remaining nodes W_{XY} as the successive application of MBO/Spectral clustering algorithms require non negative eigenvalues for stability and guaranteed convergence.

In theory a symmetric positive semi-definite matrix is guaranteed non negative eigenvalues and its eigendecomposition and singular value decompositions coincide. This is however not always the case when working numerically with real-valued data. Subroutines for many popular programming libraries (e.g. numpy and C++ LAPACK) differ between divide-and-conquer and regular QR approaches for their respective SVD and eigendecompositions. The study by Nagatsukasa et al [58] compares stability of different solvers and generally found the divide-and-conquer solvers to be both more efficient and more stable than traditional QR solvers. When running experiments the Nyström approximation would be prone to crashes with regular eigendecomposition as it would at times output slightly negative trailing eigenvalues due to rounding errors. Instead adopting SVD for all decompositions had a positive impact as it in the experiments guaranteed non negative singular values for intermediate steps 8 and 11 of algorithm 1.

The second problem seemed to stem especially from matrix inverses and the long chain of matrix multiplications and calculations in algorithm 1. Relatively small rounding errors from floating point numbers can have a large impact when accumulated over many steps and possibly ruin the final result. For a standard implementation of Nyström the smallest eigenvalues of the approximated graph Laplacian for landmark nodes $L \leq 100$ when negative mostly ranged between 1×10^{-15} to 1×10^{-16} which can be reasonably considered within the margin of error and did not seem to negatively affect the results when rounded down to 0. The error however increased with the number of landmark nodes up to 1×10^{-3} for $L > 400$, likely because of W_{XX} being more critical for the eigenvector/-value computations than W_{XY} and thus contributing more to error.

The authors of [43] needed to add a small value $\epsilon = 0.1$ to input images to ensure numerical stability when using a cosine distance measure. As mentioned there were no problems in constructing valid submatrices of W, potential problems with numerical stabiity arose from the long chain of matrix multiplications of Nyström. Something analogous to the added ϵ was considered in adding a small ridge to $W_{XX} + I_L \cdot \epsilon$ as was done by [59]. Other articles also

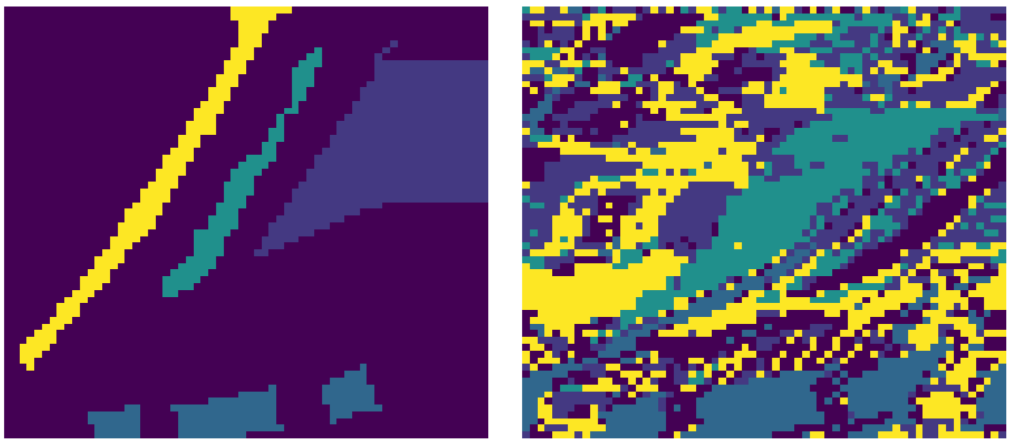
consider different ways of improving stability for calculations of the inverse in Nyström and Nyström-like algorithms by avenues such as truncation of the pseudoinverse by discarding summands of SVD [60][61], which can be especially helpful for ill-conditioned matrices. Considering the first approach this may not be ideal as the added ridge is essentially adding meaningless features to $W_X X$. In practice for small ϵ there was no noticeable degradation of results, but neither no improvement to stability for $\epsilon < 1 \times 10^{-15}$. The added ridge needed to be around $\epsilon > 1 \times 10^{-4}$ for any meaningful improvement to stability, which render results useless as the added ridge would severely alter results.

For $N = 4096$ nodes the average run time when calculating the decomposition of the true graph Laplacian was $26.579\,\mathrm{s} \pm 0.775\,\mathrm{s}$. With the same number of nodes but solving eigenvectors/-values of the graph Laplacian with the Nyström approximation average run time decreased to $0.214\,\mathrm{s} \pm 0.024\,\mathrm{s}$. Use of the Nyström method therefore resulted in a quite significant decrease in computing time, which could likely be improved upon even further by optimizing the code through vectorization and parallelization.

Observed advantages of this method was severely decreased computational cost, but as a result of this it was also possible to run graph MBO in cases where computations of the square terms of the graph Laplacian would otherwise be impossible with a regular solver since number of nodes is equal to the squared number of pixels. In ideal cases it can represent the true node space well enough to generate good predictions during diffusions without the need to calculate the full decomposituion. It can be run on entirely randomly picked landmark nodes which is reportedly often enough[13] but this did not generate adequate results in the experiments.

When landmark nodes are handpicked it also generally require very small amounts of semi-supervised data ($5-10\%$). A possible drawback of Nyström is therefore that some datasets requires landmark nodes from a balanced draw across all classes to give good results. For many real world datasets without a fully labeled groundtruth this will not be possible. Figure 4.4 shows an image where the predicted MBO classification map from randomly drawn Nyström landmark nodes has inadequately recreated the space spanned by the true nodespace.

Cases were the Nyström method seemed to struggle especially were in datasets with large numbers of classes such as the Houston set, in accordance with [13]. This can be explained in part by the coordinate change $a^n = H^T u^n$ which discards large amounts of data and when trying to project the original data into a space spanned by only a set of approximated leading eigenvectors from



(a) Ground truth. (b) GMBO segmentation mask.

Figure 4.4: Figure showing ground truth (a) and Graph MBO prediction mask (b) for a Trento data sample. Landmark nodes are here drawn randomly and this image was chosen to illustrate a case where the choice of randomly selected landmark nodes has severely failed to project the full data onto a space spanned by the eigenvectors approximated from random landmark node resulting in a heavily skewed, distorted and noisy prediction mask.

the landmark nodes this is dependent on all classes being well represented in the approximation.

Furthermore a challenge can be that other than through inspection of the segmentation map assessing the quality of different Nyström runs compared to the eigendecomposition of the true graph Laplacian is hard without actually performing calculations involving the full squared weighted graph matrix W. When the node space recreation of Nyström is highly inadequate this might result in severely skewed segmentation maps with characteristics that make it possible to conclude Nyström is the bottleneck in a classifier as was discussed with figure 4.4, but assessing this through inspection will likely often not be feasible if the errors are not as severe and/or typical for a badly projected eigenspace.

4.6.3 Segmentation masks and evaluation metrics

In theorems 5.3 and 4.2 of [62] the authors show that not all small choices for dt are valid. For too small timesteps the MBO model will not propagate label probabilities at all and remains stationary at its initialization. The diffusion will not be sufficient to change nodes neighboring the semisupervised

input nodes and the algorithm will not learn. If the timestep is too large however not only does the next iterate for the heat equation degrade because of inability to accurately approximate the mean curvature of but the MBO will reach a stationary state after only one iteration resulting in very little propagation of class probabilities and therefore a segmentation mask very close to the initialized values. In practice this was solved by choosing a timestep dt of 0.1 ± 0.3 as changes within this range gave nearly identical results (when accounting for randomness of each run), while deviation below above this range would yield random segmentation maps with convergence after one run. Going below this range gave very slow or no convergence with inadequate results.

Figures 4.5 (a) and (b) shows ground truths and predicted segmentation masks from the Trento datasample. Performance of the sample image was representative for the Trento set. The overall shapes of ground truth objects are generally correctly placed but with slight segmentation artifacts in borders between objects. The shapes and space is not particularly skewed or thwarted which could indicate that the Nyström approximation performed well and that the artifacts between object borders are more likely to stem from the MBO convergence.

The Houston segmentation map in figure 4.5 (c) and (d) has gotten a slight part of the road class cluster correct but not all and with significant bleed from the road class into its surroundings. Some of the road class is correctly placed and much of the background is correctly predicted but there is significant amounts of wrongly predicted pixels leaking from the road into the background. Class clusters in the Houston set differ from all other sets by ranging from only one to a couple of pixels wide, which seems to have negatively impacted the MBO algorithm as most noise often happens between borders. Other influencing factors are likely to be the large number of classes in this set which was found to be an issue for both reconstruction of the nodespace from eigenvectors in Nyström and the MBO in [13].

An example of the GMBO used on Svalbard NE in figure 4.6 (a) and (b) has achieved a segmentation mask very close to the ground truth. MBO seems to propagate outwards from the fidelity input and here again most noise happens between object borders, except for two smaller incorrectly labeled background clusters on the bottom right side. This could come from a bad random initialization in MBO or from those points being more similar than other parts around them to the background class. Perhaps most importantly is that the main shapes of ice is identified. If used for locating ice regions from a small amount of fidelity further analysis by human experts could be

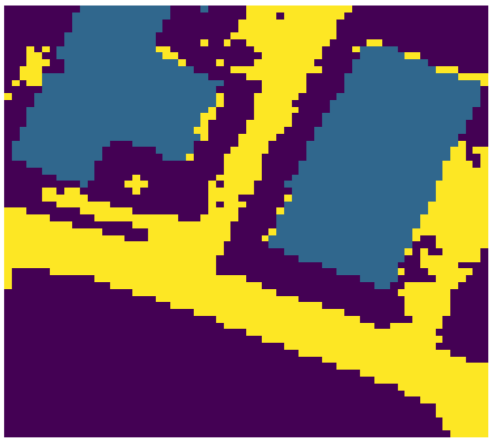
considered in determining the classes of the two smaller erronous clusters if no ground truth were available for those areas.

Sample ground truths and segmentation masks for the last dataset S Svalbard are shown in respectively (c) and (d) of figure 4.6. The phenomena here seemed especially interesting for its lack of uncertainty except between class borders and odd indent in the middle of the image. It would seem there could be a transitional landscape between the classses, perhaps with thinner and/or melting ice that blends closely with landscape in mode similarities. An explanation for the indentation could be due to MBO misclassification but considering this is only a smaller snippet from an extremely large satellite image it is also thinkable that the indentation is closer to the real landscape than ground truth and that ground truth was roughly placed relatively to the size of the image patch.

A table for comparing accuracies and mean IoU metrics for the Trento, Houston, NE Svalbard and S Svalbard datasets classified with Graph Fusion MBO is shown in table 4.3. Tables 4.4 and 4.5 respectively show class IoU scores for Trento and for Houston. Houston had the significantly higher accuracy of 0.991 next to Trentos accuracy of 0.707, while Trento outperformed Houston in Mean IoU with a mean IoU of 0.462 compared to 0.137 of Trento. The class IoU tables of Houston show an especially high class IoU for background and otherwise a trend of very poor classification scores for all other classes.

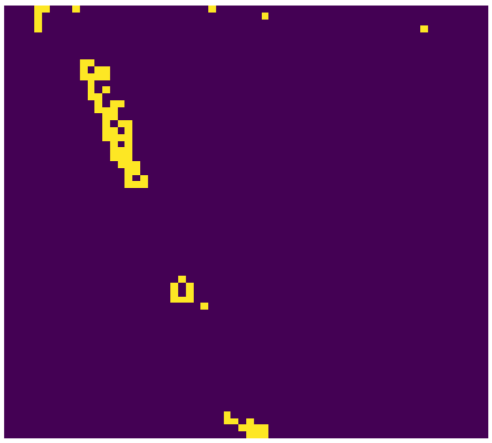
When evaluating performance on the different datasets one needs to consider the overall goal of the method used. While the Graph Fusion MBO scored higher in accuracy on the Houston dataset than on Trento, this can be explained by the majority of images from the Houston set showing background with a few scattered pixels of ground objects. Most of the background were correctly classified but very few of object classes were correctly identified. Furthermore, the data seem to show that a possible strength of the Graph Fusion MBO algorithm seems to lie in being able to indicate presence of class objects from a relatively small number of labeled fidelity input. In Trento were MBO worked especially well this seemed to be the case but a possible drawback was the artifacting and noise especially in the borders between objects, so for finer segmentation masks neural nets could be a better choice at the cost of more labeled data. Taking this into account correctly classifying large amounts of background would therefore most often come as a secondary concern in Remote Sensing tasks compared to indicating presence of object classes, as this is what Graph Fusion MBO would likely be used for in RS. The results from Trento could therefore be thought of as intuitively more valuable than results from the Houston set.

(a) Ground truth.



(b) GMBO segmentation mask.

(c) Ground truth.



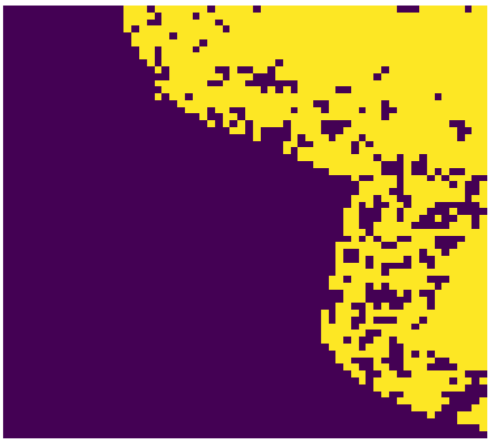
(d) GMBO segmentation mask.

Figure 4.5: A figure showing ground truths and their corresponding segmentation masks for Graph Fusion MBO in sample data from the Trento (a), (b) and Houston (c), (d) datasets. Trento shows background in dark blue, houses in light blue and road in yellow. In Houston background is shown in dark blue and road in yellow.

(a) Ground truth.

(b) GMBO segmentation mask.

(c) Ground truth.



(d) GMBO segmentation mask.

Figure 4.6: A figure showing ground truths and their corresponding segmentation masks for Graph Fusion MBO in sample data from the NE Svalbard (a), (b) and S Svalbard (c), (d) datasets. The NE Svalbard sample shows grey-white ice in yellow and background in blue. In the S Svalbard sample background is also shown in blue and brash ice is shown in yellow.

Dataset	Accuracy	Mean IoU
Trento	0.707	0.462
Houston	0.991	0.137
NE Svalbard	0.982	0.526
S Svalbard	0.993	0.715

Table 4.3: Table showing Accuracy and Mean Intersection over Union for different remote sensing datasets classified with the Graph Fusion MBO algorithm.

The arctic sea ice datasets had the highest mean IOUs of respectively 0.526 and 0.715 for NE- and S Svalbard. These datasets had the least amount of classes and very distinct clusters which has earlier been linked to improved performance in GMBO [13]. The high scoring Mean IOUs on S Svalbard can also be affected by the small number of channels as this set had only 4 regular bands compared to ranges between 15 and 70 bands for most other sets. Earlier studies has strengthened the case for not always including additional bands or modalities as beneficially merging them does not necessarily guarantee improvements to robustness nor accuracy of analysis [9]. Furthermore it was odd that open water was the class with the simultaneously lowest and highest mean IOUs for respectively the NE Svalbard and S Svalbard datasets. A possible explanation could be the use of different sensory equipment, i.e. that the AMSR sensor or bands included in S Svalbard are more relevant to distinguishing open water than the optical and SAR combination present in NE Svalbard.

For a closing remark it should be noted that the GMBO performed surprisingly good on the sea ice datasets that in part motivated this thesis. This despite that it to the authors knowledge has not been conducted extensively studies into MBO characterization of sea ice data for these combinations of SAR, AMSR and optical modalities without LIDAR data in popular GMBO papers [13][43]. The low amount of classes and distinct clusters typical in the ice data was indeed proved quite fitting for use of the Nyström and MBO algorithms. The method had its drawbacks and might not be fitting if needing finer segmentation masks or more labeled data is available. It also struggled especially for large numbers of classes which could change results similarly to what happened with the Nyström methods inability to represent each class for the Houston set.

Class name	Class IoU score
0: Background	0.626
1: Apple trees	0.326
2: Buildings	0.437
3: Ground	0.269
4: Wood	0.758
5: Vineyards	0.398
6: Roads	0.390

Table 4.4: Classes and their belonging class IoU scores for the Trento dataset when classified with Graph Fusion MBO. The classes with the two highest scores are Wood and background, while the lowest class scores are ground and apple trees.

Class name	Class IoU score
0: Background	0.991
1: Healthy Grass	0.112
2: Stressed Grass	0.097
3: Synthetic Grass	0.000
4: Trees	0.003
5: Soil	0.000
6: Water	0.243
7: Residential	0.033
8: Commercial	0.015
9: Road	0.089
10: Highway	0.016
11: Railway	0.116
12: Parking Lot 1	0.047
13: Parking Lot 2	0.075
14: Tennis Court	0.218
15: Running Track	0.000

Table 4.5: Classes and their belonging class IoU scores for the 2013 IEEE GRSS Data Fusion Houston dataset when classified with Graph Fusion MBO. Notably the background class has a score much higher than the others at 0.991 while the second best class water has a class IoU of only 0.243. Most classes besides background had very poor performances on this dataset with class IoUs ranging between 0 and 0.01.

Class name	Class IoU score
0: Grey Ice	0.667
1: Grey-White Ice	0.460
2: Open Water	0.257
3: Thick FYI	0.363
4: Thin FYI	0.421
5: Background	0.989

Table 4.6: Classes and their belonging class IoU scores for the NE Svalbard dataset when classified with Graph Fusion MBO. Background is the highest scoring class, with Ice classes scoring significantly less.

Class name	Class IoU score
0: Background	0.972
1: Brash Ice	0.546
2: FYI	0.606
3: Open Water	0.738

Table 4.7: Classes and their belonging class IoU scores for the S Svalbard dataset when classified with Graph Fusion MBO. This set only consisted of four classes but their IoUs were were mostly higher than the other sets, with the largest after background being 0.738 for Open Water.

Chapter 5

Conclusions and next steps

The motivations behind this work were to evaluate Semi-Supervised methods used successfully in other research communities together with inclusion of additionally extracted GLCM texture feature data in the analysis. Deep Label propagation and Holistic MixMatch were SSL methods that considered processing of multimodal datasets simultaneously for image classification objectives. The last method, Graph Fusion MBO, looked into the parallel handling of modalities with graph fusion at a later stage with a focus on spectral graph image segmentation. Each method was experimentally evaluated on real world arctic and urban datasets to provide an exhaustive description of their respective drawbacks and advantages for operational use.

Out of LP and MixMatch the former were the lowest scoring algorithm which coincided with the original hypothesis of MixMatch being a more powerful algorithm since LP is somewhat outdated. Baseline LP had error rates of 13.54 and 10.95 while MixMatch had 9.39 and 5.40 for 120 and 240 labeled samples. Generally LP had error rates between 10-14, with little dependence on tested hyperparamters except for the L2 normalization which was an important part of LP. LP was also highly dependent on phase 1 generating learned parameters representative of class. It is possible that a different set of hyperparameters, dataset or number of epochs trained in phase 1 would decrease the difference between error rates of LP and MixMatch. MixMatch was more sensitive to changes in amount of labeled samples, and hyperparameters than LP. The most important component of MixMatch was MixUp.

If the experiments for image classification in this project, i.e. Label propagation and MixMatch, are to be recreated, the following would ideally be done differently/added.

- Larger dataset: A challenge in remote sensing is the lack of well labeled data, and a larger more varied dataset could be beneficial to model performance. Clusters of the NE Svalbard combined dataset were very distinct, and at most 3-4 larger clusters existed per class. The dataset should be large and varied enough such that the model is able to learn a well generalized representation of data to classify new samples. When only trained on one area the trained models are likely to face some overfitting, and might not work well if used on data from a different location, or with less distinctly separated class clusters.
- Other architectures: Concatenated mixed input data fed into WideResNets or ConvNets might not be the best models for classifying multimodal satellite image data. A possible alternative for mixed data is sending different modes in parallell networks, e.g. ConvNets for S1 and S2 data and a Multilayer perceptron for GLCM, and concatenating learned features at a later stage with learned weighting based on the importance of each mode.
- More finely tuned hyperparameters: This is a timeconsuming process, but especially LP could have benefitted from this considering how important the first phase was.

A continuation for future work with LP and MixMatch could be attempting pixel-wise segmentation with these methods. This would be of value as it would be more true to the original dataset as it is labeled per pixel, and would open for comparisons with GMBO. For ice classification there could also be practical cases requiring higher precision where an image wise classification is not enough. There is no guarantee that these image classification algorithms will work well when used directly in semantic segmentation. This could be hypothesised for LP as it was found to be the lowest performing method in this study, and could be considered somewhat outdated. MixMatch used for semantic segmentation would however be especially interesting as the original authors states an interest in it being used for other domains, as well as in hybrid with other SoA methods.

Another possibility is comparing the currently implemented SSL algorithms to more state-of-the-art algorithms, especially since LP is not a SoA approach and introducing more algorithms could be better suited for comparing with MixMatch, as well as performing better on multimodal satellite images. Particularly Graph Convolutional Networks for Hyperspectral Image Classification [63] and FixMatch [27]. GCN for Hyperspectral Image Classification would be a good comparison for LP as a more recent graph based state of the art method. FixMatch was created by many of the authors of MixMatch and

consistently scored higher than MixMatch and other SoA on the commonly used benchmarks Cifar10, Cifar100 and SVHN.

The performance of the Graph Fusion MBO algorithm varied significantly depending on the dataset. GMBO struggled especially on the Houston set with a mean IOU of only 0.137 which could be explained in part due to the large number of bands needed to be condensed down by a fusing operation with inherent information-loss and a large amount of classes that both Nyström and MBO struggles to represent from a smaller number of landmark nodes and labeled data. For the datasets where it performed better such as Trento, with a mean IOU of 0.462, it still generated coarse segmentation maps in the regions between objects and as such might not be suitable for all use cases depending on how fine the segmentation map needs to be. Potential uses are more likely to involve object detection, as it can have merit in indicating classes given quite small amounts of sampled data $5 - 10\%$ and when the Nyström method was able to adequately recreate the graph node space from approximated eigenvectors it decreased the computation time drastically and also allowed computations for graphs larger than what would be feasible using calculations involving the true Graph Laplacian. The best performance was achieved for the sea ice datasets where NE Svalbard had a mean IOU of 0.526 and S Svalbard had 0.715. Experimental results seemed to indicate that their inherent clustering properties and fewer number of classes is ideal for GMBO.

For the GMBO algorithm thoughts about further work and recreating the experiments includes:

- Implementation of other methods for comparison purposes. Graph Induced Learning on subspaces[45] is more closely related to GMBO and seeks to better utilize the higher quality modality over others by aligning a shared subspace graph. The Graph Convolutional Network of [64] trains spectral filters of a deep learning model but still goes under the scope of spectral graph based methods for a slightly different perspective.

- A further look into the stability of the Nyström method. The long chain of matrix operations could induce large errors starting from small rounding errors, and the methods aimed at tackling this as well as computations of the pseudoinverse for PSD matrices from other papers mainly included their own flaws.

- Ablation studies targeting choice of GMBO hyperparameters. This process was made difficult due to the stability issues discussed above

when the number of landmark nodes increased, and would be more attainable after stability is improved. Some hyperparameters did not seem to change results much as long as they were within a given range but this should be looked into again after stability is further improved.

- More datasets. More thorough testing can always be beneficial to evaluating an algorithms robustness when facing data of different statistical and physical properties. In short term this would likely mean gathering sea ice data with 10 or above classes to see if an increase of classes in sea ice data leads to as sharp of a degrade in results as for Houston would be an interesting continuation.
- Other hyperspectral modality fusion procedures, including other choices of similarity measures for the currently implemented fusion module.

Chapter 6

Bibliography

Bibliography

[1] Leonid Polyak, Richard Alley, John Andrews, J. Brigham-Grette, Thomas Cronin, Dennis Darby, Arthur Dyke, Joan Fitzpatrick, Svend Funder, Marika Holland, Anne Jennings, Gifford Miller, Matt O'Regan, James Savelle, Mark Serreze, Kristen St. John, James White, and Eric Wolff. History of sea ice in the arctic. *Quaternary Science Reviews*, pages 1757–1778, 01 2010.

[2] Stein Sandven, Ola M. Johannessen, and Kjell Kloster. *Sea Ice Monitoring by Remote Sensing*, chapter 2, pages 1–5. American Cancer Society, 2006.

[3] Paulo Tavares, Norma Beltrão, Ulisses Guimarães, and A. Teodoro. Integration of sentinel-1 and sentinel-2 for classification and lulc mapping in the urban area of belém, eastern brazilian amazon. *Sensors*, 19, 03 2019.

[4] Lars-Anders Breivik, Tom Carrieres, Steinar Eastwood, A.H. Fleming, Fanny Girard-Ardhuin, J. Karvonen, Ronald Kwok, W. Meier, Marko Mäkynen, Leif Pedersen, Stein Sandven, Markku Simila, and Rasmus Tonboe. Remote sensing of sea ice. In *OceanObs'09 Community White Paper*, 01 2009.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

[7] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang,

Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3, 2019.

[8] Philippe Thomas. Semi-supervised learning by olivier chapelle, bernhard schölkopf, and alexander zien (review). *IEEE Transactions on Neural Networks*, 20:542, 01 2009.

[9] S. Chlaily, M. D. Mura, J. Chanussot, C. Jutten, P. Gamba, and A. Marinoni. Capacity and limits of multimodal remote sensing: Theoretical aspects and automatic information theory-based image selection. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–21, 2020.

[10] Fariba Khoshghalbvash and Jean X. Gao. Integrating heterogeneous datasets by using multimodal deep learning. In Qilian Liang, Xin Liu, Zhenyu Na, Wei Wang, Jiasong Mu, and Baoju Zhang, editors, *Communications, Signal Processing, and Systems*, pages 279–285, Singapore, 2020. Springer Singapore.

[11] Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 76:036106, 10 2007.

[12] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning, 2019.

[13] G. Iyer, J. Chanussot, and A. L. Bertozzi. A graph-based approach for data fusion and segmentation of multimodal images. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–11, 2020.

[14] J.B. Campbell and R.H. Wynne. *Introduction to Remote Sensing, Fifth Edition*. Guilford Publications, 2011.

[15] Filsa Bioresita, Anne Puissant, André Stumpf, and Jean-Philippe Malet. Active and Passive Remote Sensing Data Time Series for Flood Detection and Surface Water Mapping. In *EGU General Assembly Conference Abstracts*, EGU General Assembly Conference Abstracts, page 10082, April 2017.

[16] Mryka Hall-Beyer. Glcm texture: A tutorial v. 3.0 march 2017, 03 2017.

[17] Osvaldo Simeone. A very brief introduction to machine learning with applications to communication systems. *CoRR*, abs/1808.02342, 2018.

[18] Michael W. Berry, Azlinah Mohamed, and Bee Wah Yap, editors. *Supervised and Unsupervised Learning for Data Science*. Springer International Publishing, 2020.

[19] Yuxi Li. Deep reinforcement learning. *CoRR*, abs/1810.06339, 2018.

[20] Jesper E. van Engelen and H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109:373–440, 2019.

[21] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[22] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng Ma, Xiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 311–327, Cham, 2018. Springer International Publishing.

[23] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning, 2020.

[24] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning, 2019.

[25] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Olkopf. Learning with local and global consistency. *Advances in Neural Information Processing Systems 16*, 16, 03 2004.

[26] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017.

[27] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence, 2020.

[28] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring, 2020.

[29] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D. Cubuk, and Ian J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms, 2019.

[30] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning, 2016.

[31] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks, 2020.

[32] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning, 2017.

[33] Charles H. Martin and Michael W. Mahoney. Traditional and heavy-tailed self regularization in neural network models, 2019.

[34] Ahmet Iscen. LabelProp-SSDL: Label Propagation for Deep Semi-supervised Learning. `https://github.com/ahmetius/LP-DeepSSL`, 2019.

[35] Yui. MixMatch: An unofficial PyTorch implementation of MixMatch: A Holistic Approach to Semi-Supervised Learning. `https://github.com/YU1ut/MixMatch-pytorch`, 2020.

[36] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017.

[37] Geoffrey S. Iyer. *Graph-Based Data Fusion Methods*. PhD thesis, University of California, 2018.

[38] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering, 2016.

[39] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning, 2018.

[40] Andrea Bertozzi and Arjuna Flenner. Diffuse interface models on graphs for classification of high dimensional data. *SIAM Review*, 58:293–328, 01 2016.

[41] Ekaterina Merkurjev, Tijana Kostić, and Andrea Bertozzi. An mbo scheme on graphs for classification and image processing. *SIAM Journal on Imaging Sciences [electronic only]*, 6, 10 2013.

[42] C. Garcia-Cardona, E. Merkurjev, A. L. Bertozzi, A. Flenner, and A. G. Percus. Multiclass data segmentation using diffuse interface methods on graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1600–1613, 2014.

[43] Zhaoyi Meng, Ekaterina Merkurjev, Alice Koniges, and Andrea Bertozzi. Hyperspectral image classification using graph clustering methods. *Image Processing On Line*, 7:218–245, 08 2017.

[44] Ekaterina Merkurjev, Andrea Bertozzi, Xiaoran Yan, and Kristina Lerman. Modified cheeger and ratio cut methods using the ginzburg–landau functional for classification of high-dimensional data. *Inverse Problems*, 33:074003, 07 2017.

[45] Danfeng Hong, Jian Kang, Naoto Yokoya, and Jocelyn Chanussot. Graph-induced aligned learning on subspaces for hyperspectral and multispectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):4407–4418, 2021.

[46] Xin-Ye Li and Li-jie Guo. Constructing affinity matrix in spectral clustering based on neighbor propagation. *Neurocomputing*, 97:125–130, 11 2012.

[47] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[48] Ulrike von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

[49] O. Goldschmidt and D. S. Hochbaum. Polynomial algorithm for the k-cut problem. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 444–451, 1988.

[50] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[51] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrddottextom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence - PAMI*, 26, 01 2004.

[52] Serge Belongie, Charless Fowlkes, Fan Chung, and Jitendra Malik. Spectral partitioning with indefinite kernels using the nyström extension. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision — ECCV 2002*, pages 531–542, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[53] Yves Gennip and Andrea Bertozzi. Gamma-convergence of graph ginzburg-landau functionals. *Advances in Differential Equations*, 17, 04 2012.

[54] C. A. Coburn and A. C. B. Roberts. A multiscale texture analysis procedure for improved forest stand classification. *International Journal of Remote Sensing*, 25(20):4287–4308, 2004.

[55] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

[56] Kaveri Chatra, Venkatanareshbabu Kuppili, and Damodar Edla. Texture image classification using deep neural network and binary dragon fly optimization with a novel fitness function. *Wireless Personal Communications*, 108, 05 2019.

[57] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, Oct 2018.

[58] Yuji Nakatsukasa and Nicholas Higham. Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the svd. *SIAM Journal on Scientific Computing*, 35, 01 2013.

[59] Zhihua Zhang. The matrix ridge approximation: Algorithms and applications. *Machine Learning*, 97, 12 2013.

[60] E. Izquierdo and V. Guerra-Ones. Numerical stability of nystrom extension for image segmentation. In *Proceedings of the 2004 14th IEEE Signal Processing Society Workshop Machine Learning for Signal Processing, 2004.*, pages 609–614, 2004.

[61] Yuji Nakatsukasa. Fast and stable randomized low-rank matrix approximation, 2020.

[62] Yves van Gennip, Nestor Guillen, Braxton Osting, and Andrea L. Bertozzi. Mean curvature, threshold dynamics, and phase field theory on finite graphs. *Milan Journal of Mathematics*, 82(1):3–65, Apr 2014.

[63] Danfeng Hong, Lianru Gao, Jing Yao, Bing Zhang, Antonio Plaza, and Jocelyn Chanussot. Graph convolutional networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–13, 2020.

[64] Song Ouyang and Yansheng Li. Combining deep semantic segmentation network and graph convolutional neural network for semantic segmentation of remote sensing imagery. *Remote Sensing*, 13:119, 12 2020.