FUSING IMAGE AND NON-GRID-LIKE DATA FOR OBJECT SEGMENTATION

Lappeenranta-Lahti University of Technology LUT
Master's Program in Computational Engineering

Brno University of Technology (BUT)
Faculty of Information Technology

2024

Bc. Samuel Repka

Examiners: Associate Professor Tuomas Eerola
Professor Pavel Zemčík

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Master's Thesis Assignment

157990

Institut:	Department of Computer Graphics and Multimedia (DCGM)
Student:	**Repka Samuel, Bc.**
Programme:	Information Technology and Artificial Intelligence
Specialization:	Computer Vision
Title:	**Fusing image and non-grid-like data for object segmentation**
Category:	Computer vision
Academic year:	2023/24

Assignment:

1. Study literature sources and existing solutions for fusion of image (raster image, presumably RGB) and non-image data, such as depth map, point cloud, or spectral data. Study also solutions/tasks researched at LUT and BUT e.g. in collaboration with companies.
2. Propose a method of fusion and a suitable task of the image and non-image data, such as detection of objects or segmentation in order to get results difficult or impossible to get from single modality, such as detection of knots on trunk/clade, segmentation based on image and spectra, detection of material properties, etc.
3. Investigate the capabilities and features of the proposed method and selected task and a suitable way of demonstration of the results and their metrics on a data set available at LUT or BUT (or a newly collected data).
4. Implement the proposed methods and demonstrate the results on a suitable task, discuss the features and achievable parameters.
5. Present and discuss the results and possible continuation of the work.

Literature:
According to the instruction of the supervisor and/or consultants.

Requirements for the semestral defence:
1-3

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Zemčík Pavel, prof. Dr. Ing., dr. h. c.**
Consultants:	Reich Bořek, Ing. Tuomas Eerola, Fedor Zolotarev
Head of Department:	Černocký Jan, prof. Dr. Ing.
Beginning of work:	1.11.2023
Submission deadline:	25.5.2024
Approval date:	22.5.2024

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Computational Engineering

Bc. Samuel Repka

Fusing image and non-grid-like data for object segmentation

Master's thesis
2024
69 pages, 43 figures, 1 table, 0 appendices
Examiners: Associate Professor Tuomas Eerola and Professor Pavel Zemčík

Keywords: computer vision, multimodal data, data fusion, image segmentation

Quite often, a phenomenon of interest can described by more than one data source. For example, a car's appearance shows its colour and brand, but not its engine status. However, other data sources do provide us with this information, be it a sound or mere touch. Such data source is often referred to as a modality. While using a single data source to extract the needed information may be sufficient, the addition of more modalities can be beneficial, because of their complementary nature. This data fusion, however, may be a quite challenging process. Different kinds of data have different properties, structures and various challenges connected to them. A plethora of different methods has been proposed, but usually, the methods are very data-dependent. This thesis presents a new approach to the fusion of two modalities, primarily for the purpose of image segmentation. One of the modalities is image, and the second one is any non-grid-like modality. The method uses a graph to jointly represent both modalities, aiming to capture the intra and inter-modalities relationships as accurately as possible. The graph is then processed, producing a graph with fused data, or a direct segmentation. The proposed method was evaluated on two datasets (from the fields of mineralogy and timber processing) and compared to another solution, showing both the potential and limitations of the method. In case of the mineralogy dataset, the results are very encouraging, showing that the method is capable of data fusion, even outperforming a contemporary method. In case of the timber dataset, the results were not as conclusive, as the method failed to improve the results when compared to a baseline solution, which may have been caused by a challenging dataset.

ROZŠÍRENÝ ABSTRAKT

Dáta z rôznych dátových zdrojov popisujú rôzne vlastnosti sledovaného subjektu. Napríklad, pohľad na auto dokáže odhaliť jeho farbu, značku a podobne, ale až sluch alebo dotyk dokáže povedať, či motor beží. Tento príklad ilustruje, že mať viac zdrojov dát sa môže oplatiť, pretože rôzne zdroje môžu poskytovať komplementárne informácie. Spájanie dát, tiež označované ako dátová fúzia, ale nie je práve triviálna záležitosť. Existuje obrovské množstvo zdrojov, formátov, významov dát, čo do veľkej miery komplikuje tvorbu všeobecného postupu na dátovú fúziu. Kvôli komplexnosti úlohy sú postupy často veľmi závislé na konkrétnych typoch dát a ťažko sa adaptujú na iné domény. Cieľom tejto práce je navrhnúť postup, ktorý bude pomerne všeobecne aplikovateľný na fúziu dvoch typov dát: obrazových a iných, ktoré sú bližšie nešpecifikované a primárne neštrukturované do mriežky.

Neštrukturovanosť dát sa dá popísať ako vzorkovanie objektu, ktoré nie je v pravidelnej mriežke, ako napríklad zhluky bodov alebo merania s náhodnými vzorkovacími miestami. Táto vlastnosť zvyšuje úroveň obtiažnosti, keďže zamedzuje priamemu použitiu veľkého množstva postupov, ako sú napríklad konvolučné neurónové siete alebo MLP (Multi-Layer Perceptron). Neznalosť presného typu dát tiež znamená, že vyvinutá metóda musí byť extra flexibilná, aby sa dokázala adaptovať na čo najväčšie množstvo formátov dát.

Finálny výstup metódy má byť sémantická segmentácia obrazu, teda jeho delenie na významovo podobné oblasti. Navrhnutý postup modeluje vstupné dáta jedným spoločným grafom, úlohou ktorého je čo najpresnejšie zachytiť vzťahy medzi dátovými bodmi, ktoré sú rôzneho typu a zároveň aj medzi dátovými bodmi z rovnakého zdroja. Tento graf je potom spracovaný grafovou neurálnou sieťou, ktorej cieľom je, voliteľne, buď vyprodukovať nový graf so spojenými dátami, alebo priamo segmentáciu. V záverečnej fáze sa extrahuje časť grafu reprezentujúca obraz, ktorá predstavuje výsledok spracovania obrazu. Navrhnutý postup obsahuje aj voliteľné bloky primárne určené na extrakciu relevantných dátových čŕt v prípade, že samotná grafová neurálna sieť nedokáže poskytnúť požadovaný výsledok, alebo na redukciu počtu dimenzií, ak je potrebná.

Riešenie tejto práce bolo overené na dvoch nezávislých datasetoch. Prvý test prebehol na datasete z oblasti mineralógie, vytvorený skenovacím elektrónovým mikroskopom. Obrazová časť dát bola tvorená snímanými spätne odrazenými elektrónmi a neštrukturované dáta EDS (Energy-Dispersive X-ray Spectroscopy) spektrami, ktoré presne popisujú chemické zloženie minerálu. Cieľom bolo segmentovať obraz na oblasti s rovnakým chemickým

zložením, spolu aj s určením, o aký minerál ide. Tento experiment dopadol veľmi úspešne a jasne ukázal, že navrhnutá metóda je schopná dátovej fúzie a dosahuje dobré výsledky aj s veľmi malým množstvom dát. Tiež prebehlo porovnanie s už existujúcou metódou, ktoré ukázalo, že navrhnutá metóda dosahuje lepšie výsledky vo všetkých metrikách v takmer všetkých prípadoch. Zároveň vzhľadom na dataset a spôsob trénovania ale nedokáže tak dobre generalizovať.

Druhý test prebehol na datasete drevených brvien. Dátové zdroje v tomto prípade boli fotografia celého obvodu dreva, spolu so vzhľadom k rozlíšeniu obrazu riedkymi výškovými meraniami povrchu dreva laserovným skenerom. Cieľom bolo určiť, kde sa nachádzajú uzly (pozostatky po konároch v kmeni) na brvne. Aplikovanie navrhnutej metódy ale neprinieslo želaný efekt, pretože nedokázalo vylepšiť predikciu vzhľadom k referenčnému riešeniu a ani referenčné riešenie samotné. Nie je úplne známe, či neúspech bol spôsobený nevhodnou metódou alebo samotným datasetom, ktorý obsahuje pomerne malé množstvo dát, čo veľmi sťažilo trénovanie riešenia.

Celkovo práca ukazuje, že navrhnuté riešenie má potenciál a dokáže dosiahnuť veľmi dobré výsledky, aj keď má svoje limitácie.

Kapitola 1 obsahuje úvod do problematiky. Nasledovaná je kapitolami popisujúcimi jednotlivé aspekty problematiky, konkrétne kapitola 2 uvádza multimodálne dáta, kapitola 3 neurónové siete a dátové štruktúry. Časť práce popisujúcu súčasné poznatky uzatvára kapitola 4 o dátovej fúzii. V kapitole 5 je popísané vytvorené riešenie, ktorého experimentálne overenie je uvedené v kapitole 6. Práca je ukončená diskusiou v kapitole 7 a záverom v kapitole 8.

PREHLÁSENIE

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením v zložení: Assoc. Prof. Tuomas Eerola, Prof. Pavel Zemčík, D.Sc. Fedor Zolotarev a Ing. Bořek Reich. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

. . . . . .

Bc. Samuel Repka

24. mája 2024

ACKNOWLEDGEMENTS

I would like to thank my supervisors prof. Dr. Ing. Pavel Zemčík dr. h. c., assoc. prof. Tuomas Eerola, Ing. Bořek Reich, Fedor Zolotarev, D.Sc., for their insight and guidance while working on this thesis. Also, I would like to thank my family, girlfriend and friends for their support and patience, for they were there for me when I needed it. My gratitude also belongs to Ing. David Motl from TESCAN GROUP, a.s. for his invaluable help with this thesis. Furthermore, I would like to thank RNDr. Karel Breiter, DSc., (Institute of Geology of the Czech Academy of Sciences) and Mgr. Marek Dosbaba (TESCAN GROUP, a.s.) for providing the mineralogy data. Last but not least, I am grateful to Finnos Oy for providing the timber dataset.

I acknowledge usage of ChatGPT 3.5 for idea researching and Grammarly for grammar checking.

As a wise Librarian once said: *"Ook"*.

Lappeenranta, May 23, 2024

Bc. Samuel Repka

LIST OF ABBREVIATIONS

AUROC	Area Under the Receiver Operating Characteristic
BSE	Backscattered Electrons
CNN	Convolutional Neural Network
CT	Computed Tomography
DBN	Deep Belief Net
EDS	Energy-Dispersive X-Ray Spectroscopy
FN	False Negative
FNN	Feedforward Neural Network
FP	False Positive
GAT	Graph Attention Network
GCN	Graph Convolutional Network
GDLS	Graph-based Deep Learning Segmentation
GIN	Graph Isomorphism Network
GNN	Graph Neural Network
IoU	Intersection over Union
kNN	k-Nearest Neighbours
MLP	Multi-Layer Perceptron
MRI	Magnetic Resonance Imaging
NN	Neural Network
RBM	Restricted Boltzmann Machine
RGB-D	RGB + depth
ROC	Receiver Operating Characteristic
RSCNN	Relation-Shape Convolutional Neural Network
SEM	Scanning Electron Microscope
SFCNN	Spherical Fractal Convolutional Neural Network
SLP	Single-Layer Perceptron
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
YOLO	You Only Look Once

CONTENTS

1 Introduction

1.1 Background

Object segmentation is a computer vision task, where the goal is to split an image into different regions, where each region represents an area of interest or object of interest. For example, let us assume that the task at hand is to find animals in the image. A solution could be assigning labels to each pixel, where the label would express which animal this pixel is part of, if any. Neighbouring pixels with the same labels would then together create a region representing an animal, showing where exactly it is in the image. It has various applications, e.g. in industrial, microscopy, and traffic monitoring domains, such as defect detection, material analysis, and vehicle tracking. The goal of this thesis is to improve contemporary segmentation methods by adding another data source.

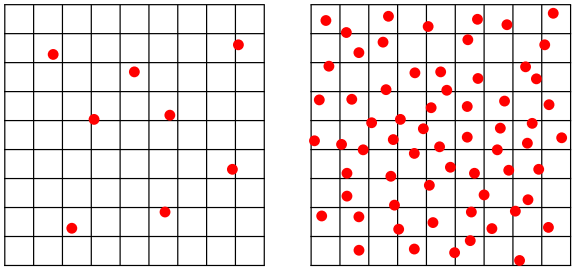
While segmenting pure RGB images has achieved great success, image data alone may not sufficiently capture all relevant information. For example, dark parts of the image can hide a relevant object, or similar colouring of the background and an object can camouflage the object, leading to incorrect results. Therefore, using another data modality to improve the segmentation accuracy [1] [2] has been shown to be a viable approach.

Data fusion is a process of integrating data from multiple sources (modalities) to leverage source-specific information carried by said data. Examples of such modalities include RGB image + depth data [3] [4], RGB and thermal data [5], and many others. Despite its complex definition, data fusion is something we all do every day without even realizing it. Our brains fuse data from our senses constantly. For example, seeing a car in front of us provides information about a car's position, while sounds carry information about whether the car's engine is running or not.

Even though the task of data fusion is easy for biological beings, when it comes to computational systems, the situation changes. Contemporary approaches are heavily dependent on the data types being fused, their structure, the overall goal of the fusion, and the context in general [6].

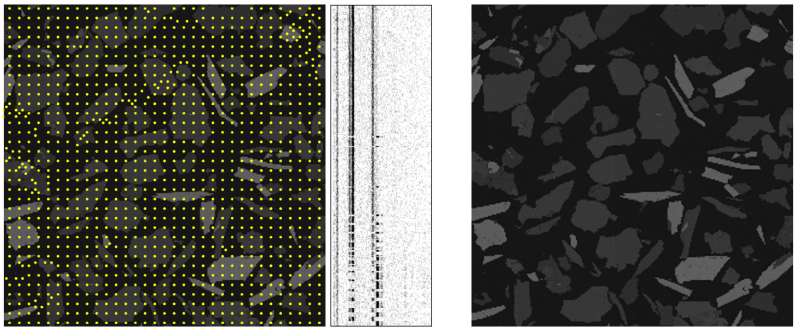
Moreover, data fusion is commonly applied on grid-like data, where all modalities are concatenated along the channel dimension and processed together, or an encoder-decoder network with two encoders is utilized. However, not every modality has a grid-like structure, for example, sparse point-wise spectral measurements (Figure 2) or point clouds

(Figure 3). Approaches to fusing such modalities must be adapted to the processing of data with diverse structures. This is usually done by using a modality-specific neural network architecture, which confines methods to only work with given data modalities. This work aims to create a general fusion pipeline easily adaptable to any unstructured data.



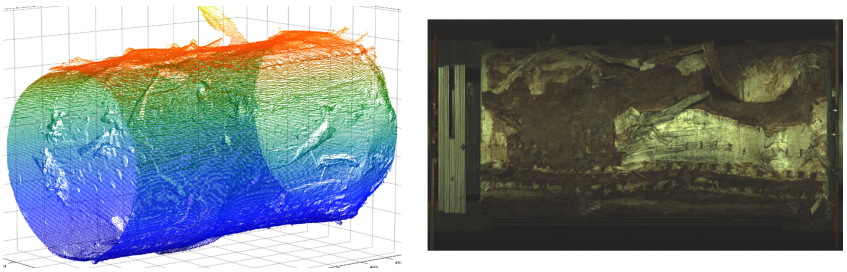
(a) (b)

Figure 1. Grid-like data (raster image) and unstructured data (red): (a) sparse unstructured data (e.g. spectral measurements); (b) dense unstructured data (e.g. point clouds). Images from an ongoing project.



(a) (b)

Figure 2. Example of the microscopy data: (a) spectral measurements on top of a microscopy image: (b) BSE (microscopy) image. Images from an ongoing project.



(a) (b)

Figure 3. Example of the sawmill data: (a) laser point cloud log representation; (b) image of a log captured by a line camera. Images from an ongoing project.

1.2 Objectives and delimitations

This Master's thesis work will focus on developing multimodal fusion techniques to improve the segmentation accuracy over the methods utilizing just a single modality (RGB image). The applications and modality pairs to be considered in the thesis include:

1. mineralogy segmentation from microscopy images: greyscale images and pointwise spectral measurements (Figure 2)
2. knot segmentation (sawmill industry): RGB images and point clouds (Figure 3)

The main objectives of the work are as follows:

1. Propose a method to fuse unstructured data to image data to obtain a new (grid-like) representation encompassing data from both modalities.
2. Implement an image segmentation method capable of utilizing the fused representation as input.
3. Compare the multimodal segmentation method to other segmentation methods on the selected applications.

While the method will be tested on the aforementioned modality pairs, the solution should be more general, and applicable on numerous other modality pairs where image is one of the modalities.

1.3 Structure of the thesis

The document is structured as follows:
In Chapter 2, different kinds of multimodal data are presented, sources of this data as well as different problems that arise from the great variability inherent to such data. Neural networks relevant to this thesis are introduced in Chapter 3, together with examples of architectures suitable for working with different data types. Chapter 4 describes data segmentation itself and how is it connected to data fusion, different approaches, and state of the art in data fusion for image segmentation. Described are methods utilizing deep learning, together with examples and references to practical applications of said approaches.

The proposed solution can be found in Chapter 5, which is followed by a detailed description of the implementation and experiments in Chapter 6. This chapter also includes the descriptions of the datasets on which the implementation was tested. In Chapter 7 an analysis of the solution can be found, with a retrospective to the objectives set in Chapter 1. The thesis is concluded with Chapter 8.

2 Multimodal data

Information about the environment, processes, or phenomena are sensed and described using detectors. Whether of biological nature or artificial, all sensors provide some kind of data that carries information about the observed. Multiple sensors can be employed for observation of the same phenomenon, for example, light and sound or RGB + depth (RGB-D) data. Moreover, each such detector may exhibit different properties based on the environmental conditions, observation times, and so on. Each such data source can be referred to as a *modality* [6].

2.1 Multimodal data description

Why are data sources important? A single modality rarely provides a complete picture of the observed process. Additional modalities may provide information unavailable in the original modality, thus improving the performance of desired tasks like segmentation or classification. For example, an RGB image carries information about the colour, texture, and general appearance of the scene. While the geometry of the scene can be estimated from such an image, it is a non-trivial task and requires assumptions or additional information about the scene [7]. However, when RGB data are paired with depth data, geometry data are directly accessible without any computation and additional information necessary. This additional modality can for example reveal that a single-coloured area on the image hides an object, or that on the contrary there is no object even though it seems like there is. An example of such data can be seen in Figure 4 [8]. Such datasets were already successfully used, and various methods [3] [9] showed that using additional modality boosts the performance of the segmentation.

With multiple data sources, the question of how to process the data, in a way that exploits information from all available modalities, arises. This process of *data fusion* is often non-trivial and use-case dependent. More on that in Chapter 4.

Multimodal data were successfully used in numerous fields for a variety of tasks. Examples of such concrete applications are, among others, in:

- **Medicine** - Shomorony *et al*. [10] used modalities like metabolome, microbiome, genetics, and advanced imaging to identify novel biomarkers and disease signatures.

(a) RGB image (b) Depth image

Figure 4. Example of RGB and depth data images [8]. Irregular illumination on the chair may cause incorrect segmentation, but the depth image does not have such defects.

- **Meteorology** - Boussioux *et al*. [11] created an approach combining spatio–temporal data with statistical data for hurricane forecasting.
- **Robotics** - Zhang *et al*. [12] utilized RGB, thermal, and point cloud data modalities to safely realize human-to-robot object handover.
- **Mineralogy** - Juranek *et al*. [1] used a backscattered electron image fused with energy dispersive X-ray spectroscopy data for mineral phase analysis.

These examples and many more show that multimodal data can provide a more complete and accurate description of the observed, and exploiting the complementarity can be beneficial for various tasks.

2.2 Challenges of multimodal data

Multimodal data exhibits a diverse array of characteristics, because of various sensor properties and measurement environments. This variability is the cause of many issues, that need to be considered with working with such data. In [13] and [6], the following challenges are listed.

Non-commensurability

Multimodal data are often not directly comparable because they describe different properties of the object or scene of interest. For example, earlier mentioned RGB data report on the intensity of light in specific wavelengths, while depth data describe how far the object is. Comparing those values by value would probably not lead to any meaningful result, thus the data modalities are non-commensurable [13] [6].

Noise

Because no measurement tool has absolute precision, noise is ever-present in all datasets. Because of the multimodality, an additional challenge lies in the fact that data from different modalities may contain noise of different types or properties. The reason for this is that different measurement tools can have different precisions, due to operating under various conditions, calibrations, or just because they are not made for the same precisions [13] [6]. Depending on the chosen fusion strategy, the heterogeneous nature of the noise may need to be accounted for, although most works do ignore this kind of challenge.

Missing or misaligned Data

Data might not be present where it is expected to be, or the value is unreliable. Reasons for this are varied and can be caused by one of three main reasons [13] [6].

1. A missing sample is caused by an unreliable/faulty detector.
2. Modalities cannot cover the same parts of the object of interest. As an illustration, RGB data covers only the surface of the object, while an additional modality, such as an X-ray scan, covers the whole volume. As such, the RGB data inside the object can be considered missing.
3. Sampling of the modalities is incomparable. The cause of this can be a different resolution or non-uniform sampling, as illustrated in Figure 1.

Conflicting, contradicting or inconsistent Data

Since multiple data sources are used for a task, it is possible that the information contained in each of the modalities is conflicting. Caused by an incorrect measurement, external interference, or something else, it may need to be taken into account. This may or may not be a problem depending on the data fusion method chosen. For example, if modalities are processed independently and fused are just results, a conflict resolution method may be necessary (for example by voting) [13] [6].

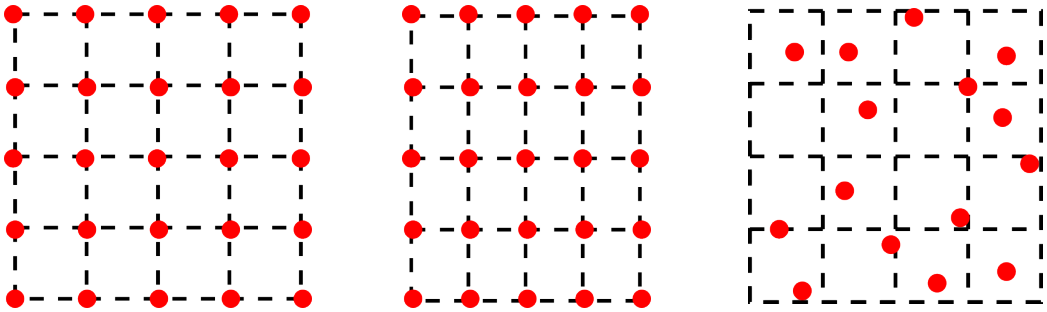
Different structure

The structure of the modalities can be extremely varied. Variations in structure include different numbers of dimensions (such as RGB and depth data), different resolutions (e.g. RGB + spectral measurements), and a completely different structure of modalities (such as RGB and point clouds). Handling such a diverse array of structures is one of the main challenges of working with multimodal data and one of the reasons why there is no one-fit-all solution for data fusion [13] [6].

It is also important to clarify, what data are considered to be *structured* or *grid-like*, and what data are not. When referring to structured data, data sampling points must be defined on a *regular grid*. A regular grid is defined as follows [14]:

> *"The grid is regular if its cells are identical rectangular prisms (bricks) aligned with the axes."*

Any other data, that potentially do not conform to this definition, are considered to be unstructured. Illustration of grid-like and non-grid-like data can be seen in Figure 5.



(a) Grid-like data (b) Still grid-like data (c) Non-grid-like data

Figure 5. Examples of grid-like and non-grid-like data. The red dots are sampling points.

3 Deep Learning

This chapter introduces neural networks and various architectures made for specific data types. It is by no means a comprehensive overview. It is intended to be a general introduction and description of architectures relevant to this thesis. The full description of the topic is not possible because of the limited scope of the work.

3.1 Neural networks introduction

Neural networks are a class of machine learning algorithms. They are incredibly powerful tools for modelling, forecasting, and classification, with the most basic one being a *Feedforward Neural Network (FNN)*. The goal of such a network is to find a function f such as $\boldsymbol{y} = f(\boldsymbol{x}, \boldsymbol{\theta})$, where $\boldsymbol{x}$ is input data and $\boldsymbol{y}$ is the output, for example, a category for a classification task [15]. $\boldsymbol{\theta}$ is a set of learnable parameters, which the network learns by training.

The term feedforward comes from the fact that in such a network data flows always forward, meaning there are no feedback connections (such a network would be recurrent) [15]. FNNs are networks because the function f is often composed of multiple functions, for example, as follows:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = f^{(1)}(f^{(2)}(f^{(3)}(\boldsymbol{x}, \boldsymbol{\theta^{(3)}}), \boldsymbol{\theta^{(2)}}), \boldsymbol{\theta^{(1)}}) \tag{1}$$

Each of the sub-functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$ is called a layer. The network represented by the equation is illustrated in Figure 6.

The first layer is called the visible (or input) layer. Subsequent layers except for the very last one are often called hidden layers. The final layer produces the output of the whole network and as such is called the output layer [15]. The network can be represented by a computational graph, which must be directed and acyclic for feedforward networks, similar to Figure 6, which further highlights the notion of the "network".

The simplest type of a FNN is a Multi-Layer Perceptron (MLP). In order to understand, what an MLP is, it is important to introduce the concept of a single perceptron. A perceptron is a simplified mathematical model inspired by the basic functioning of a neuron. The workings of a perceptron are described in literature [16] as follows:

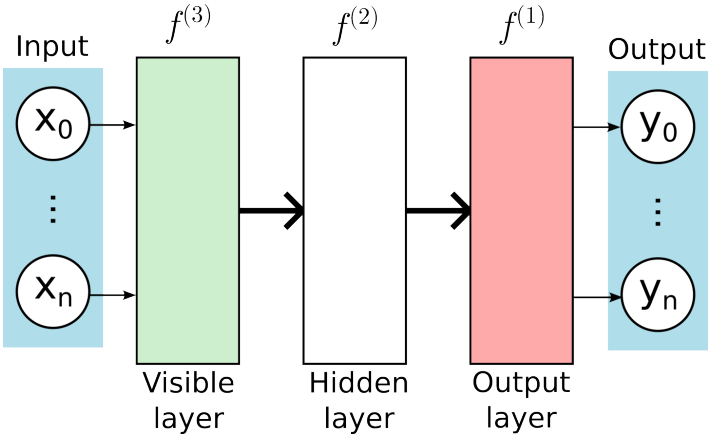


Figure 6. Illustration of data flow in feedforward neural network with respect to Equation 1.

$$y = f_a(\sum_{i=0}^{n} w_i x_i + b) \tag{2}$$

where y is the output of the perceptron, x_i and w_i are inputs and corresponding weights ranging from 0 to n, and b is the bias term. The function f_a is an activation function, introducing non-linearity into the term (a simplified representation can be seen in Figure 7a). Parallel connection of the perceptrons creates a Single-Layer Perceptron (SLP) (Figure 7b), which is also a synonym for a **fully connected layer** (also known as the **linear** or **dense** layer). Stacking of SLPs creates a MLP architecture, which is an example of a FNN. With respect to Equation 1, SLP layers correspond to the sub-functions $f^{(i)}$, while MLP correspond to the final term $f(\boldsymbol{x}, \boldsymbol{\theta})$ [16]. A graphical illustration can be seen in Figure 7c.

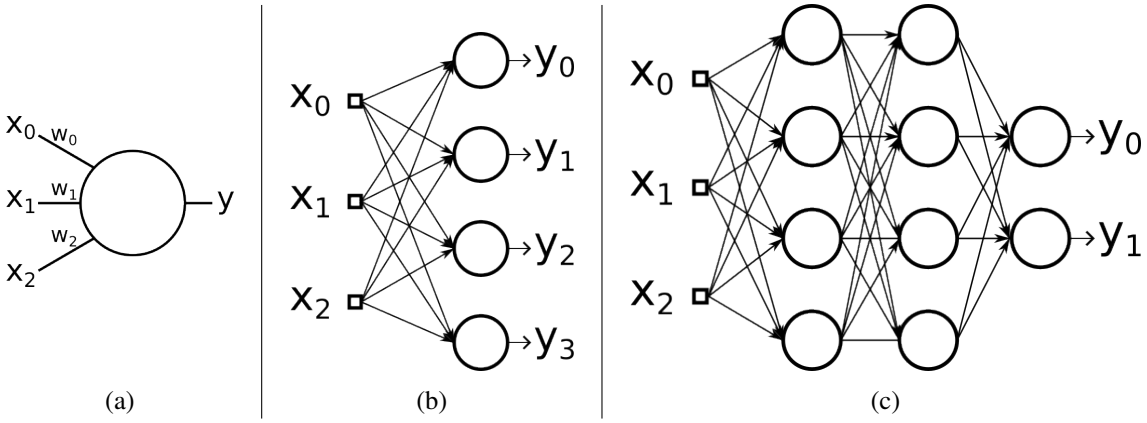


Figure 7. Illustration of a perceptron (a), SLP (b), and MLP (c). Adapted from [16].

The neural network has to be trained in order to produce reasonable results. The goal of

the training is to find values of $\boldsymbol{\theta}$ in such a way that maximizes the similarity of $f(\boldsymbol{x}, \boldsymbol{\theta})$ output $\boldsymbol{y}$ with required output $\boldsymbol{y*}$ [15]. The problem of similarity maximization is often formulated as minimization of error. In order to calculate the error, a loss function must be defined, such as mean squared error for regression tasks or categorical cross-entropy for classification tasks. The problem of finding the optimal parameters θ^* can be then formulated as:

$$\theta^* = \arg\min_{\theta} \sum_{i=0}^{N} loss(f(x_i, \theta), y_i) \tag{3}$$

where N is the size of the training dataset, x_i and y_i are input data and desired output from the dataset, respectively, and $loss$ is a loss function of choice.

The training data provide rough approximations of the desired final function evaluated at different points. These samples are used in the training process, which is iterative and exploits the gradient of the function to estimate the best values for function parameters $\boldsymbol{\theta}$. The most used algorithm for estimating the function gradients is called *back-propagation* [15]. The back-propagation computes the chain rule of calculus while traversing the neural network in a backward manner, which is much more efficient than the evaluation of the analytical expression for the network gradient. After the gradients are calculated, the parameters θ are updated using an algorithm such as stochastic gradient descent.

3.2 Architectures for structured data

Besides previously mentioned MLP, many other architectures suitable for structured data exist. The most widely used are Convolutional Neural Networks (CNNs) and Deep Belief Nets (DBNs). The presented examples are not exhaustive, as other architectures are appropriate for structured data as well. However, these are not relevant to this thesis, and as such, they are not described here.

Convolutional neural networks

Convolutional neural networks are perhaps the most widely used kind of neural networks for image processing. Their applications are not constrained to images, however. They are suitable for all data that have known, grid-like topology. They introduce two new types of neural layers: *convolutional* layers and *pooling* layers [15].

The **convolutional layer**, unsurprisingly, performs convolution on the data. Convolution (denoted with asterisk $*$) can be defined for 1-dimensional data as well as for multidimensional data. For example, for 2D data, the convolution is defined as follows [15]:

$$S(i,j) = (I * K)(i,j) = \sum_{m}^{h} \sum_{n}^{w} I(i-m, j-n)K(m,n) \tag{4}$$

where I are 2D data (e.g. greyscale image), K is a 2D kernel, i, j are the coordinates of a concrete data point and h and w are dimensions of the kernel. Benefits of using convolutional layers include sparse interactions, parameter sharing, or equivariant representations [15]. Convolutional layers work as feature extractors for the convolutional layers. For example, filters in the first convolutional layer of AlexNet [17] work among others as edge and corner detectors (Figure 8 [18]).

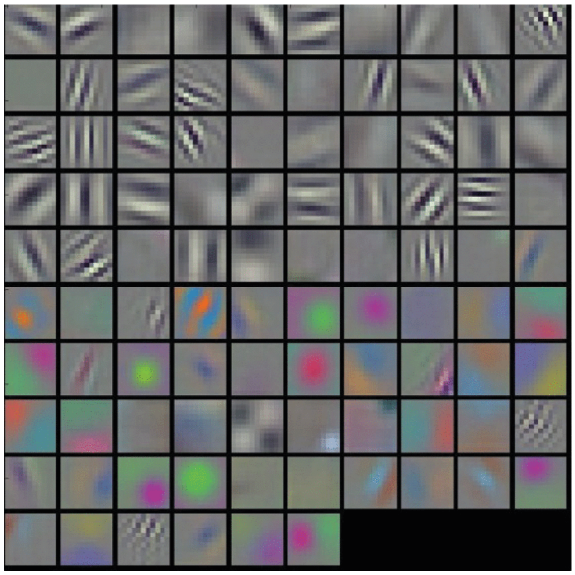


Figure 8. Visualized filters from the first layer of the AlexNet [18].

Pooling layers replace input data at a certain location in some range (for example rectangular area) with some summary statistics about it, such as maximum or mean. Pooling enables the output to be invariant to slight translations of the input, improves computational efficiency, and reduces memory requirements [15].

As is usual for neural networks, the aforementioned layers are often stacked, making a deep neural network. Various configurations of layers lead to different architectures suitable for different computer vision tasks. These include VGG-16, U-Net, or You Only Look Once (YOLO), all of which are briefly described below.

- **VGG-16** - VGG is an architecture of a neural network originally created for object classification and localisation by the Visual Geometry Group at Oxford University. Multiple variants exist, such as VGG-16 and VGG-19, where the number denotes the layer count in the network. It is composed of a series of convolutional layers with 3x3 filters, followed by pooling layers, with fully connected classification layers at the end [19]. Illustration can be seen in Figure 9 [20]. The simplicity and good performance of the architecture make it a popular choice for many machine learning tasks even today.

- **U-Net** - Originally introduced in [21] in the year 2015, U-Net is an architecture designed for biomedical image segmentation. It is composed of a contracting path (not dissimilar to a VGG), which captures context, and an expansive path, which enables the actual segmentation (assigning labels to each pixel, in original paper termed *localisation*). The architecture (illustrated in Figure 10) is symmetric and has skip connections between layers of the same size, which helps to preserve spatial information. From its inception, U-Net has been widely used in various image segmentation tasks, not only in the biomedical field.

- **YOLO** - This architecture is made for yet another computer vision task - object detection. As the name suggests, the network processes each image only once, producing bounding boxes and probabilities for each class in a single pass, which contrasts with other methods like R-CNN [22]. The network is composed of convolutional layers, with the last layer being a fully connected one, outputting the bounding boxes and probabilities [23], as depicted in Figure 11 [23]. Various versions have been created since its introduction, with the latest currently being YOLOv9 [24].

In summary, CNNs have become the de facto standard in image processing using neural networks. They excel in tasks involving visual data, such as classification, image recognition, and object detection. However, they are not restricted to computer vision tasks, as they are also widely utilized for example in natural language processing tasks [25].

Deep belief networks

The DBN is a type of neural network, which is composed of Restricted Boltzmann Machines (RBMs). An RBM is a generative stochastic neural network with the ability to learn a probability distribution over its set of inputs. RBMs have been successfully used

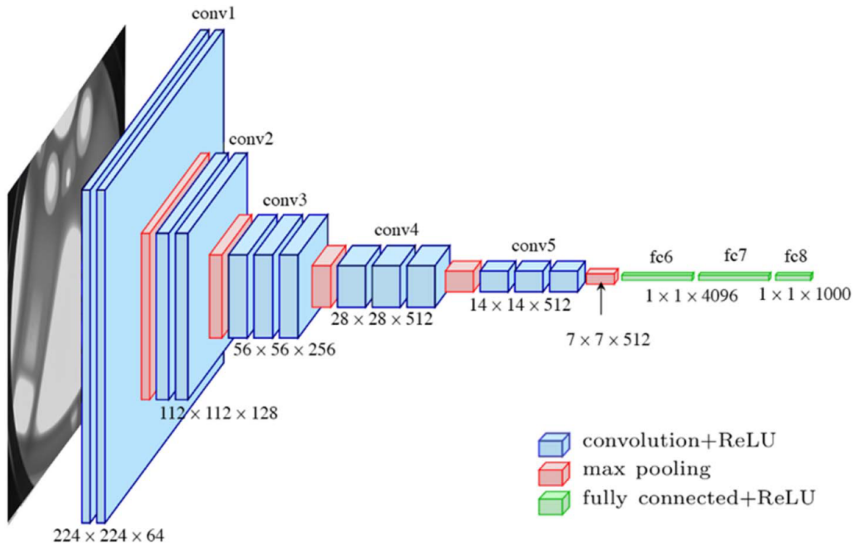


Figure 9. Example of a CNN. The architecture shown is VGG-16 introduced in [19]. The image was taken from [20].

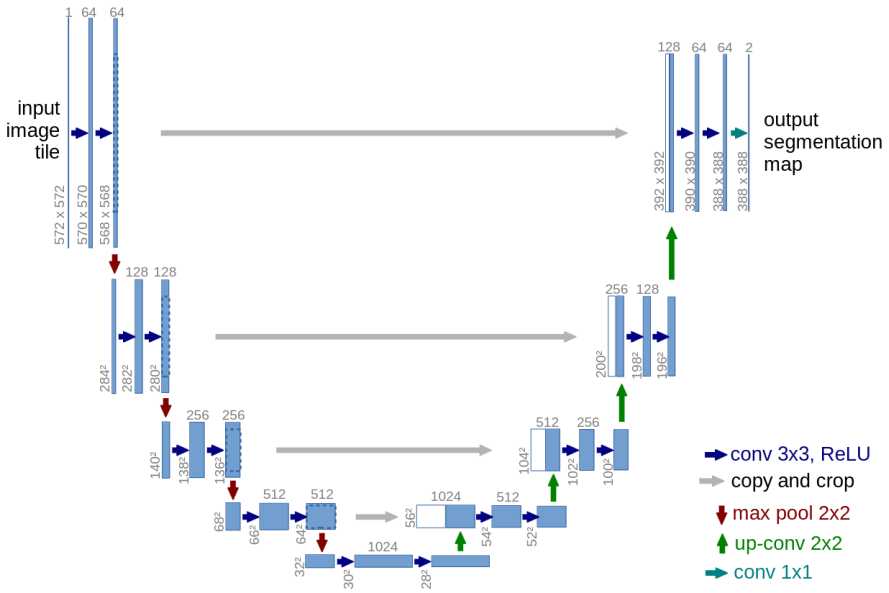


Figure 10. Illustration of the original U-Net [21].

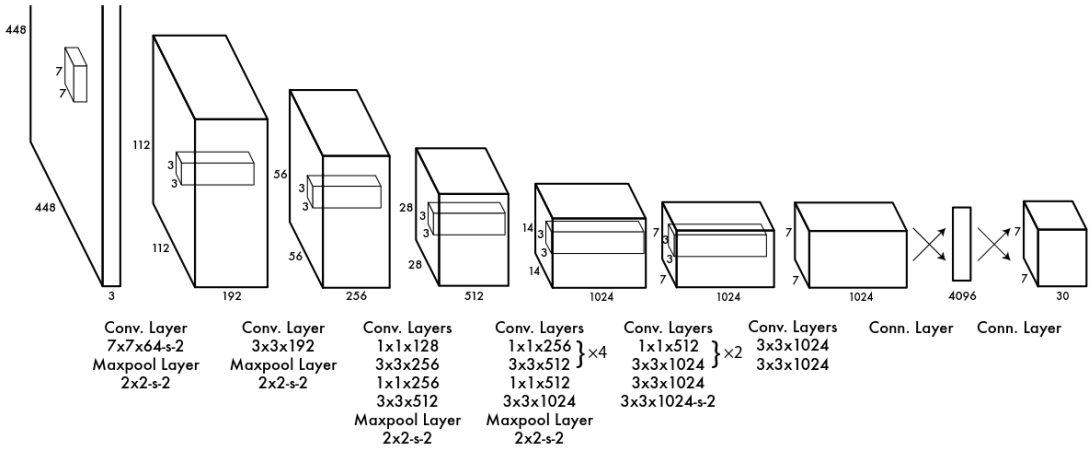


Figure 11. YOLO architecture [23].

in multiple tasks, including dimensionality reduction, feature learning, classification, and multimodal data fusion [26] [27].

An RBM is a layered model with two types of layers: visible and hidden. The layers consist of units that are fully connected between layers, however, there are no intra-layer connections. Associated with units are binary weights, which can be written in the matrix W. The matrix is of size $m \times n$, where m is the number of visible units and n is the number of hidden units. Besides weights, bias values are also associated with both layers, a for the visible layer and b for the hidden one. Contrary to the conventional neural networks, the model is trained using the *contrastive divergence* [28] method. Moreover, the model is energy-based, meaning it uses the energy function to calculate the probability distribution between the visible and hidden layers. The probability distribution P over the input x is calculated using the equation

$$P(x, h) = \frac{e^{-E(x,h)}}{Z} \tag{5}$$

where x and h are the visible and hidden layers, respectively. E is the aforementioned energy function and Z is the normalizing function, both of which are in the following forms:

$$E(x, h) = -\sum_j a_j x_j - \sum_i b_i h_i - \sum_j \sum_i h_i w_{ij} x_j \tag{6}$$

$$Z = \sum_x \sum_h e^{-E(x,h)} \tag{7}$$

where w_{ij} is the weight representing connection between, a_j and b_i are biases of visible and hidden layer respectively. Variable x_j is the value of a visible unit and h_i value of a hidden unit [29] [27].

Deep belief net is composed of such RBMs, hidden layers of the first layer being a visible layer of second RBM, hidden layer of second being a visible layer of third RBM, and so on. A graphical explanation can be seen in Figure 12.

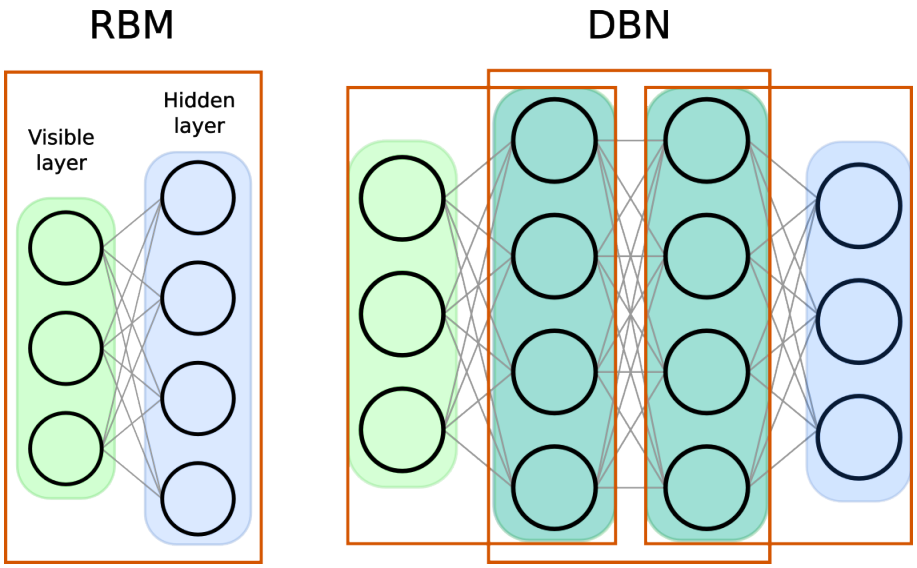


Figure 12. RBM on the left, with visible and hidden layers highlighted. DBN on the right, the orange rectangle means an instance of an RBM.

Many models utilizing DBNs have been created to date. They have been used to solve tasks like Alzheimer's disease diagnostics, human body pose estimation, face recognition, and more. DBNs are used to transfer modality-specific representations into the shared spaces, over which the joint probability distribution is modelled. While DBN based models excel at capturing the informative features, they generally lack an understanding of temporal and spatial data properties [27].

3.3 Architectures for unstructured data

Unstructured data are data that do not have a grid-like structure. Examples of such data are point clouds, graphs, or other measurements with variable sampling points. In this section, the most used approaches for processing such data using neural networks are presented. The presented examples are not exhaustive.

Conversion to structured data

One of the more obvious methods of processing unstructured data, for example, point clouds, is to transform them into some kind of structured data. This is usually done using voxelization or projection. Voxelization converts 3D data to a 3D grid, on which a 3D CNN can be applied [30] (Figure 13 [31]). This approach can be generalized to other-dimensional data, where grid-discretization is applied to the data and subsequently, the data are treated as structured data. This approach works reasonably well, however, depending on the dimensionality of the data, the memory and processing requirements scale not too well. For example, for 3D data, the requirements grow cubically.

Similar approaches that intend to alleviate this flaw utilize hierarchical structures like KD-trees and octrees to efficiently encode the input data [30].

Methods based on the projection to another structured representation are, for example, multi-view-based methods for point clouds that exploit the projection of the 3D shape into multiple views, which are subsequently used for further processing. The goal of these methods is to extract features from the views and fuse them into a global descriptor. Spherical Fractal Convolutional Neural Network (SFCNN) [32] utilizes projection onto a sphere discretized with an icosahedral lattice.

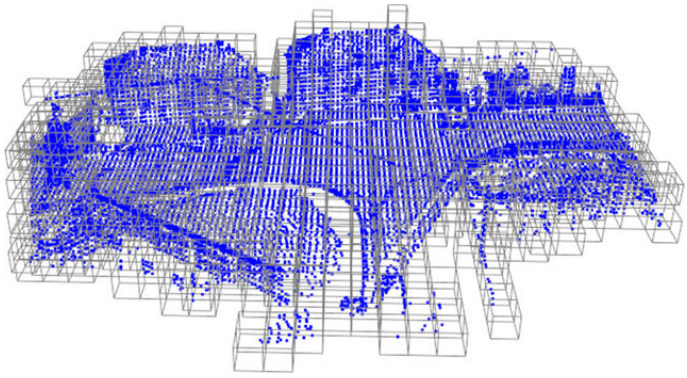


Figure 13. Voxelization of a point cloud [31].

Pointwise MLP

Neural Networks (NNs) implementing pointwise MLP are able to directly process point clouds without any preprocessing steps. Qi *et al*. [33] created the PointNet, the first NN of its kind using this approach. PointNet uses several MLPs to extract pointwise

features from each point, after which a max-pooling is used to extract global features. An illustration can be seen in Figure 14 [30]. This approach has shown to be promising, however, because of independent point processing, the local structural information was not captured. PointNet++ [34] alleviated this problem using a hierarchical variation of this network.

Many other architectures based on PointNet have been created, such as Point Attention Transformers [35], PointWeb [36], and more. An excellent overview can be found in [30].

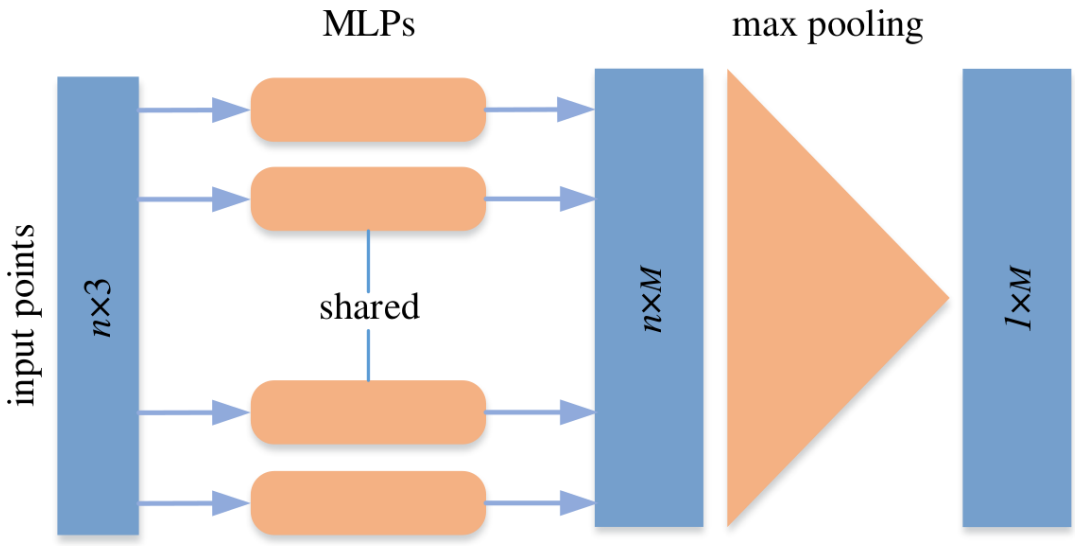
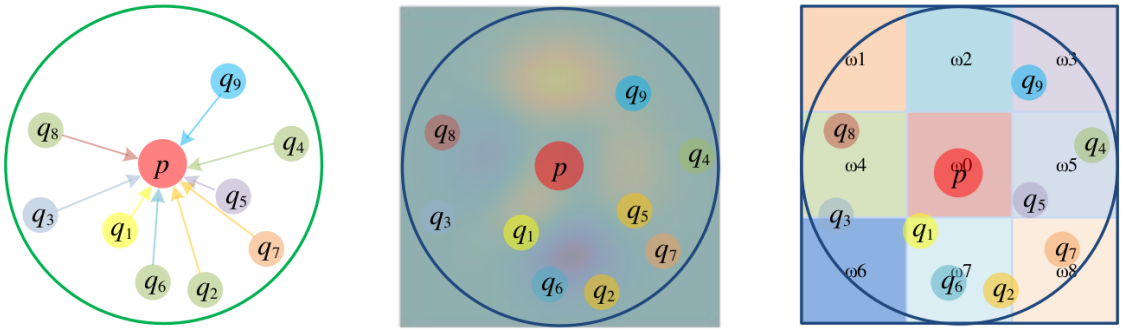


Figure 14. Simplified PointNet architecture [30]. n and M denote the number of input points and dimension of learned features, respectively.

Continuous convolution

Instead of discretizing the given space, certain methods opted for defining the convolution in continuous space, demonstrated in Figure 15 [30]. Wang *et al*. [37] proposed a parametrized continuous convolution, which spans the full continuous vector space. A similar approach created by Boulch *et al*. [38] defines continuous convolution in the unit sphere. Liu *et al*. [39] created a Relation-Shape Convolutional Neural Network (RSCNN), which uses Relation-Shape convolution to extract discriminative shape information from a spherical neighbourhood of a given centre. As per usual, many other approaches exist [30], as this field is actively being researched with encouraging results.



(a) 3D neighboring points **(b) 3D continuous convolution** **(c) 3D discrete convolution**

Figure 15. Illustration of a continuous convolution [30].

Graph Neural Network

Graphs are another type of approach for possibly unstructured data. While not necessarily unstructured (for example, the image can be looked at as a graph with edges connecting nodes representing pixels to neighbouring nodes [40]), in general, the graphs certainly can be unstructured. As such, methods have been created to process such data using neural networks.

A graph can be defined as a collection of entities (nodes or vertices) with relations between each other (edges). Both nodes and edges can have specified an embedding, representing some additional data associated with a given node or an edge. Additionally, global (master node) embeddings can be specified for information regarding the graph as a whole. Edges can be directed or undirected [40].

Graph Neural Network (GNN) applies a differentiable model (like MLP) to each of the mentioned graph components (nodes, edges, and global embeddings), transforming them into a new graph. Data integration between neighbouring nodes (or edges, but not node to edge or edge to node) is facilitated by *pooling* [40]. Similar to the convolution in CNN, for items to be pooled data are first gathered and then aggregated by a function, the simplest of which are sum or mean.

In order to combine information from different parts of the graph (e.g. edges to nodes), a *message passing layer* is used. It gathers neighbouring nodes/edges embeddings, aggregates them using the aggregate function (e.g. sum), and finally the data are passed through the update function, which can be for example another MLP [40]. An illustration of a simple GNN can be seen in Figure 16.

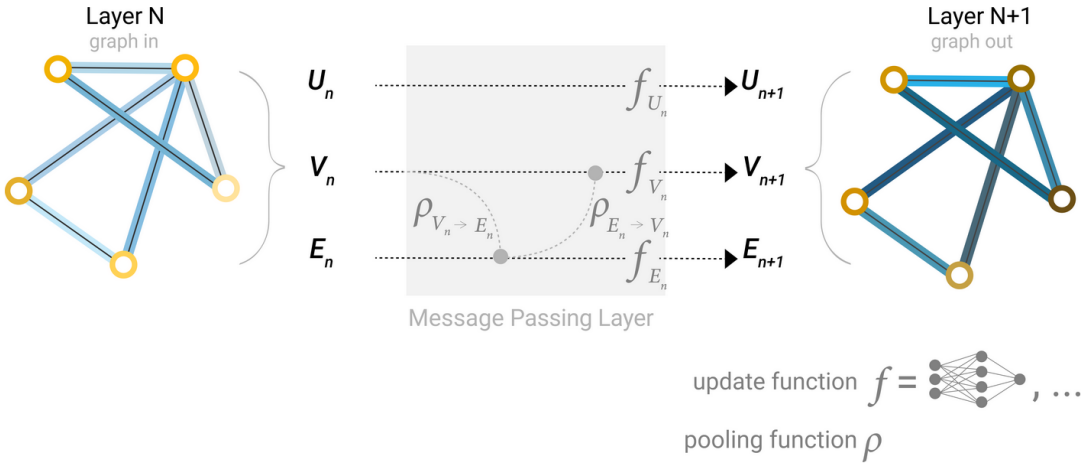


Figure 16. Example of a simple GNN [40].. U_n, V_n, and E_n represent global, node, and edge attributes at the n-th layer, respectively. In the message passing layer, vertex attributes are first pooled and then integrated with edge attributes. Subsequently, edge attributes are integrated into the vertex attributes in a similar fashion. Attributes are then passed through the update function.

Because of the high flexibility and applicability of graphs, it is not a surprise that the architectural landscape of GNNs is quite diverse. Variants include Graph Convolutional Network (GCN), Graph Attention Network (GAT) and Graph Isomorphism Network (GIN).

- **GCN** - CNNs have been proven to be a powerful tool for image processing and computer vision tasks. The GCN is an adaptation of this architecture for graph data. Introduced in [41], it uses a localized first-order approximation of spectral graph convolutions to process node features as well as a local graph structure.
- **GAT** - With the rise of attention mechanism in sequence-based tasks [42] [43], researchers started to explore the possibilities of using it in other domains and GNNs were no exception. GATs [44] use self-attention to process node neighbourhood and improve the previous spectral-based architectures. The attention mechanism allows the network to focus on the most important nodes in the neighbourhood, which can be beneficial in various tasks.
- **GIN** - While graph features (be it node, edge, or global features) are certainly important, graph structure may provide a lot of information as well. Xu *et al.* [45] introduced GIN, an architecture designed to be as powerful as the Weisfeiler-Lehman graph isomorphism test with regard to the ability to differentiate between non-isomorphic graphs.

To sum up, GNN is a powerful and versatile architecture made for the processing of

unstructured data and was successfully used in various tasks like protein folding [46], physics simulation [47], recommendation systems [48], and more [40].

3.4 Sparse data features

Another property of data that is important to mention is data sparsity. Sparse data can be both structured and unstructured. An example of sparse structured data would be for example spectral measurements with respect to an image, where spectral measurements are taken in a grid-like fashion but much more infrequently, whereas unstructured would be such data that do not follow any grid pattern, e.g. point clouds.

Sparsity in unstructured data usually does not need to be specially handled, as there is generally no notion of a missing sample since there is no reason to expect a sample somewhere in the first place. However, when considering structured sparse data, the situation changes. Traditional CNNs are ill-suited for handling sparse data, as they are inefficient and affect data in undesirable ways [49]. As such, various methods were devised that improve the efficiency of working with sparse structured data, like sparse convolutions [50] [49].

Another consideration to be made is, that structured sparse data can be transformed into a point-cloud or graph representation, which allows for the usage of different NN architectures. See, for example, [40], where it is shown that the image can be transformed into a graph representation.

4 Data fusion for segmentation

Many approaches to data fusion exist and the choice of the method is highly dependent on the data modalities, their structure, and the task at hand. This chapter aims to provide a brief introduction to the concepts of image segmentation, multimodal data fusion, and deep data fusion for segmentation. It is not an encyclopedic overview, as the size constraints do not allow it.

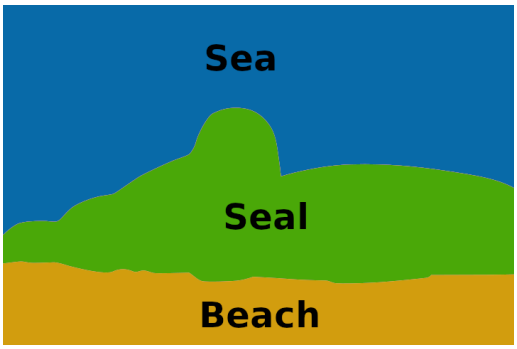
4.1 Image segmentation

One of the most important tasks in computer vision is image segmentation. It involves dividing the image into multiple parts (or segments), where each segment should have some distinct contextual significance. Segmentation can be seen as a procedure assigning each pixel a label, which conveys its semantic meaning. If the result of the segmentation is only a semantic label (for example, a car, sky, a person), the segmentation is considered to be *semantic*. If, on top of that, the segmentation partitions objects, not only the semantic labels (for example car_1, car_2, person_1, ...), it is an *instance* segmentation [51]. One can consider even the combination of the two when segmentation partitions into classes where "an instance" does not have a meaning (for example a sky), and into classes where instancing makes sense (car_1, car_2, ...). Segmentation of this type is called *panoptic* [51]. Figure 17 illustrates differences between different kinds of segmentations.

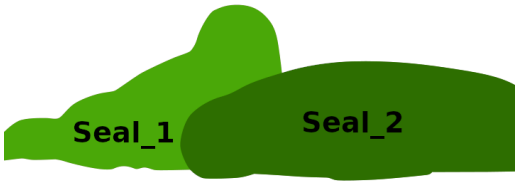
The importance of image segmentation lies in its ability to enable and/or enhance the accuracy and efficiency of subsequent tasks, such as object recognition, tracking, and scene understanding. By segmenting an image, complex visual data is organized into more manageable components, facilitating the extraction of valuable insights and information. This process is vital in various applications, including medical imaging, autonomous vehicles, satellite imagery analysis, and augmented reality [51].

Modern segmentation methods built on neural networks encompass a diverse range of techniques, such as encoder-decoder architectures (which often leverage CNNs) to progressively extract and refine features before reconstructing the segmented output. U-Net [21], already described in Section 3.2, is a widely adopted variant of this architecture. It features skip connections between the encoder and decoder parts of the network to preserve spatial information and enhance segmentation accuracy. SegNet [53], on the other hand, employs an encoder-decoder framework with a focus on efficient memory usage

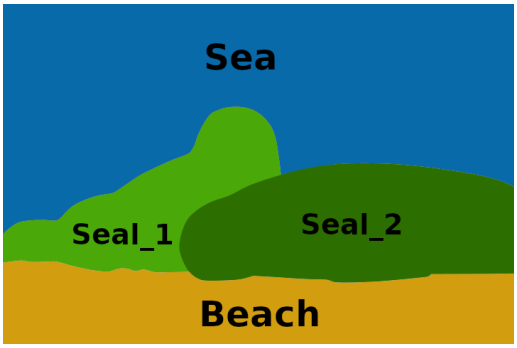
(a) Example image. Courtesy of U.S. Fish and Wildlife Service Northeast Region [52].



(b) Semantic segmentation.



(c) Instance segmentation.



(d) Panoptic segmentation.

Figure 17. Visualization of different kinds of segmentations.

through the integration of pooling indices during downsampling. Additionally, for instance segmentation, methodologies such as Mask R-CNN [54] have been devised. Mask R-CNN is built on top of Faster R-CNN (created for object detection), and extended with object mask prediction. For panoptic segmentation, an example is YOLO-panoptic [55], extending YOLOv3 with semantic and instance segmentation branches.

4.2 Multimodal data fusion

The task of multimodal data fusion has been researched since the first half of the 20th century [6]. Since then, a great number of fusion methods have been devised, approaches ranging from statistical methods [56], through tensor decompositions [6] to deep learning [27] [3] [1].

As of the current state-of-the-art, no single widely used method of fusing data exists. All approaches are very use-case dependent and developed for specific combinations of modalities, datasets with specific properties, and so on. This is due to all the chal-

lenges multimodal data contains, such as non-commensurability, variations in the structure, noise, etc [6]. More on the topic in Chapter 2.

Contemporary multimodal data fusion approaches concerning image segmentation are quite often neural-network-based. While there are some exceptions [57] [58], this chapter focuses on NN-based methods due to their success and versatility in various tasks, including image segmentation.

With data fusion arises a question: when should data be fused? Some modalities allow for direct concatenation [59] or just integration using multiplication [3] of different modalities, for example, RGB-D data. Data can be fused a bit later, after feature extraction, for example via concatenation of feature vectors [60]. Furthermore, data fusion can be in a semantic space of unimodal features, in their latent representation. Various categorisations exist for fusion methods. In this thesis, a taxonomy described in [61] is adopted.

The categorization of fusion methods according to the fusion stage is as follows:

- **Early fusion** - raw data or their features are fused, usually via concatenation
- **Late fusion** - each modality is first processed in its own branch and data fusion is performed at the decision-level
- **Hybrid fusion** - methods aiming to combine strengths of both early and late fusion methods

Visual representation of methods can be seen in Figure 18 [61]. Early fusion methods allow for optimal integration of information from different modalities, because the fact that modalities are processed together emphasizes cross-modal information interaction. Late fusion methods process modalities separately, which provides better scalability and flexibility, however, the importance of cross-modal correlation may be insufficient. Hybrid methods attempt to combine the best of both worlds, which may reflect on execution time [61].

4.3 Deep data fusion for segmentation

Data fusion has been widely applied in image segmentation tasks. Majority of contemporary methods utilize neural networks, which can be also categorized as illustrated in

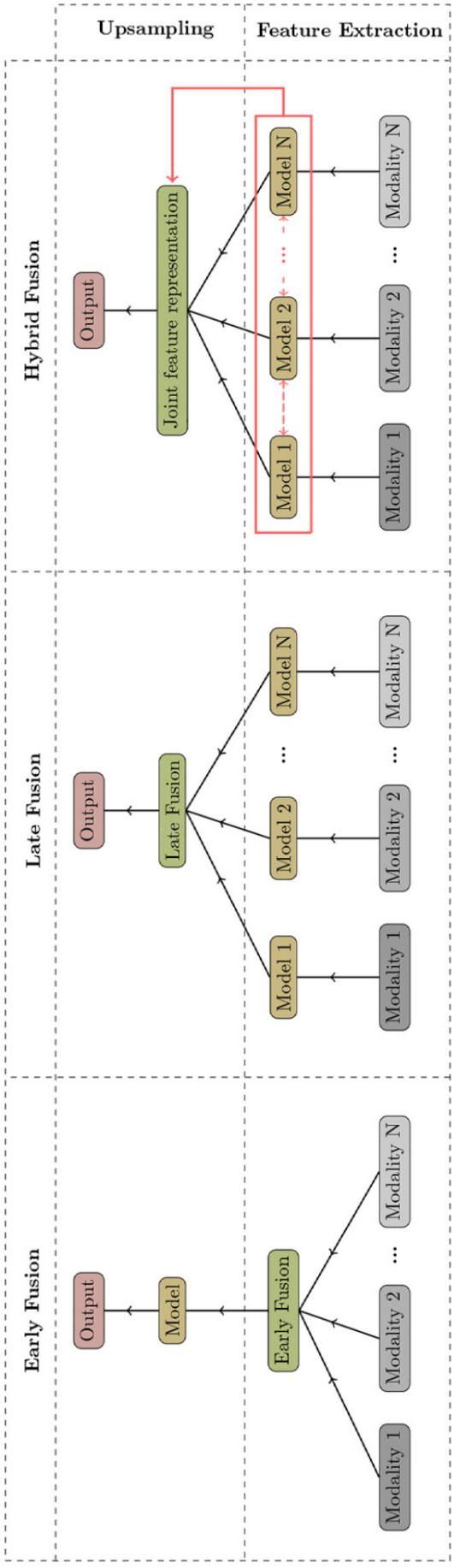


Figure 18. Visualization of different fusion methods [61].

Figure 18. The purpose of this section is to provide a brief introduction to different NN architectures for image segmentation utilizing data fusion, as well as showcase the representative models from each category.

Early fusion

In this category, data are fused in their raw form or at the feature level [61]. Raw-form fusion was utilized for example by Couprie *et al*. [62], which is the first attempt at deep multimodal fusion. Possibilities of the RGB + sparse depth data fusion using direct channel concatenation were explored by Jaritz *et al*. [4], with capability for both semantic segmentation and depth completion.

Feature-level fusion is often done by encoding both modalities separately in respective encoders, with cross-modal interactions in the encoding stage, as seen in Figure 19. Such a method is used in the FuseNet [63] with RGB-D data. RGB and thermal data fusion was explored by Sun *et al*. [5].

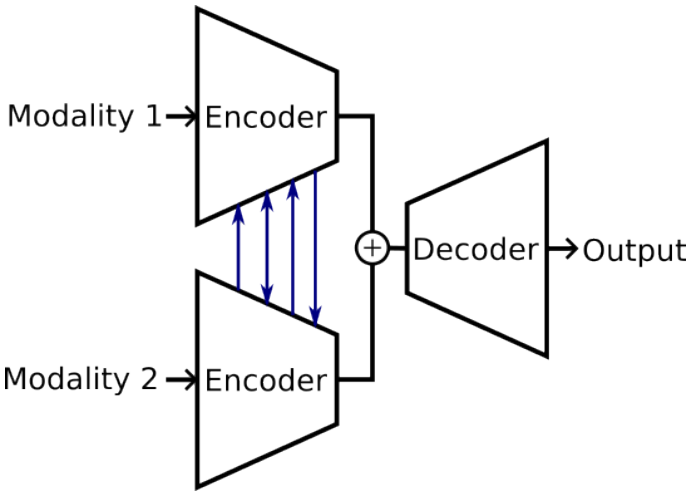


Figure 19. Illustration of one type of early fusion networks. Each modality is encoded separately, but the separation process is conditioned by cross-modality interactions (blue arrows). Encoder outputs are then fused and processed by a single decoder.

Late fusion

The paradigm of late fusion is to integrate the feature maps of modalities at the decision level. Thus, the data must be processed separately, similar to Figure 19 but without the

cross-modal interactions.

A representative sample of this category is for example work by Gupta *et al*. [64]. Two separate neural networks are used to extract features from RGB and depth data, which are combined by Support Vector Machine (SVM). PIF-Net by Guo *et al*. [65] is another example of such a network, adapted for image and point cloud fusion, capable of semantic segmentation for both input modalities.

Hybrid fusion

As mentioned previously, hybrid fusion methods are meant to alleviate the shortcomings of early and late fusion methods. Usually, skip connections are employed to bridge the encoders and a decoder, to improve the segmentation performance. An example of such architecture can be seen in Figure 20 [66], which is extracted from the article by Seungyong *et al*. showcasing the architecture of RDFNet. An alternative example can be seen in work by Fang *et al*. [67], where hybrid data fusion is used to segment brain tumours from Magnetic Resonance Imaging (MRI) modalities.

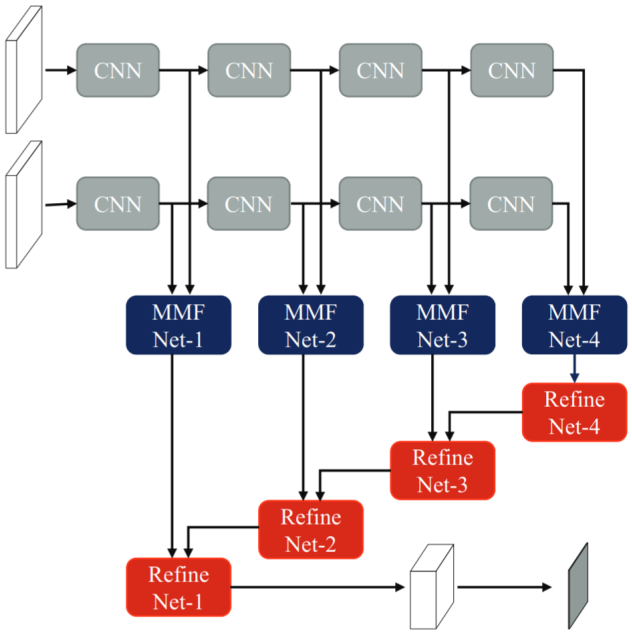


Figure 20. Example of hybrid fusion architecture. The network diagram is that of RDFNet from [66], where skip connections bridge the gap between encoders and a decoder, allowing the usage of information from all encoding stages.

5 Framework for data fusion

In this section, a general framework for the fusion of image and non-grid-like data is proposed. The framework is designed to be as general as possible, with a modular structure that allows for customization and combination with various other methods. This solution was designed to be flexible, easily expandable, and modular, in order to allow for usage of it with various modalities and for various tasks. The exploration of a new approach was also part of the motivation for this work, as to the author's knowledge this approach has not yet been tried.

5.1 Goal setting

The objective of the thesis is to create a general framework for the segmentation of an image using *image data* and *non-grid-like data*. The multitude of available methods provides a good selection of tools to choose from, however, the approach is most often very data-dependent. Since the nature of the task is very complicated, the proposed framework should focus on the following challenges:

1. The solution should be able to handle datasets with different structures.
2. Connected to the thesis is also the handling of missing values, as a dataset with missing values can be looked at as a dataset with a non-uniform structure and no missing data.
3. More often than not, the data considered here are non-commensurable. As such, non-commensurability will have to be focused on as well.

5.2 Framework description

The overview of the framework is depicted in Figure 21. It revolves around a joint representation of both modalities as a graph, and data fusion by a graph processing technique. While the graph processing step may be sufficient for the whole segmentation, experiments showed that the selected graph processing method may not have the ability to extract necessary information from the data. Because of that, the framework as such contains optional blocks for additional feature extraction or other purposes, such as dimensionality reduction. A brief description of Figure 21 follows:

- **Graph construction** - A graph is constructed from both modalities, serving as a joint intermediate representation.
- **Graph processing** - The graph is processed by a method, the task of which is to fuse data and produce either a segmentation result or fused data representation. Various approaches can be used, for example, GNNs or Graph Grammars [68]. In this work, a Graph Attention Network (GAT) was used.
- **Grid extraction** - Since the task in this thesis is image segmentation, the graph part representing the image is extracted from the graph.
- **Optional processing step** - The optional blocks may accommodate any method, that processes the respective data format. For example, in Section 6.4, an optional block is used for dimensionality reduction, or in Section 6.5, a U-Net was used to extract features from the image.

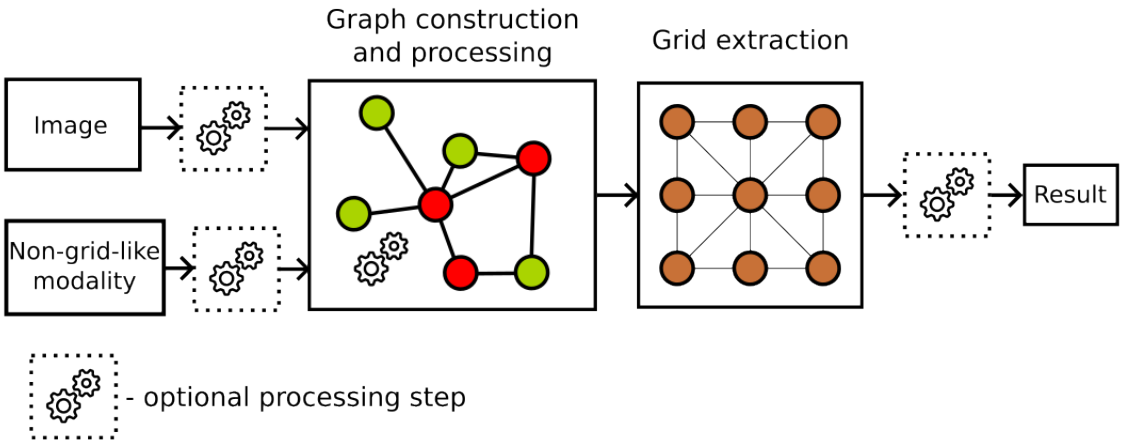


Figure 21. Proposed framework.

The framework allows for both early fusion and a variant of late fusion. If pre-graph stage optional processing blocks are substituted with feature-extraction blocks, the framework is akin to late fusion, whereas if just the post-graph stage one is used, an early fusion is performed. All optional processing blocks can be omitted, as the graph processing method in itself may be enough to extract the relevant information, as shown in Section 6.4.

5.3 Transformation to a grid-like representation

Since the final goal is image segmentation, the result of the fusion framework must be grid-like and in the same shape as the image has been. Thus, a method for the transformation of the non-grid-like data into a structured form must be devised.

Furthermore, the proposed method assumes that there exists a spatial relationship between modalities and that both modalities are aligned. For example, in the case of image and depth data, it is assumed that the depth measurements correspond to the same spatial locations as the image pixels. This alignment allows for a meaningful fusion of the two modalities, as the spatial relationships can be leveraged to improve the segmentation results. Because of these assumptions, the method presented is not meant to be applicable to modality pairs such as image and text, image and sound, etc.

There are multiple ways how to transform the non-grid-like representation into a structured form. For example, by interpolation or by aligning to the grid and filling with zeroes where data are unknown. In order to stay as general as possible, a decision was made to represent both modalities in a joint form using a graph. This allows for the exact spatial representation which also allows for "weighing" of sample points with respect to each other. This can be achieved by assigning graph edges connecting points a number expressing how likely are those measurements correlated. After the graph processing, the part originally representing the image would be extracted, resulting in a grid-like representation.

Edge construction

Edges of the graph should be constructed in such a way, which allows for the optimal leveraging of spatial relationships between data nodes. Because of the properties of modalities, the edges can be split into three subgroups:

- Edges between nodes of the image
- Edges between nodes of the second modality
- Edges between nodes of both modalities

Edges between data nodes of the image are perhaps the most simple ones. As the image is always structured, the direct solution is to create edges connecting neighbouring pixels

in 4 or 8-neighbourhood. Here, a choice was made to utilize the 8-neighbourhood, as illustrated in Figure 22.

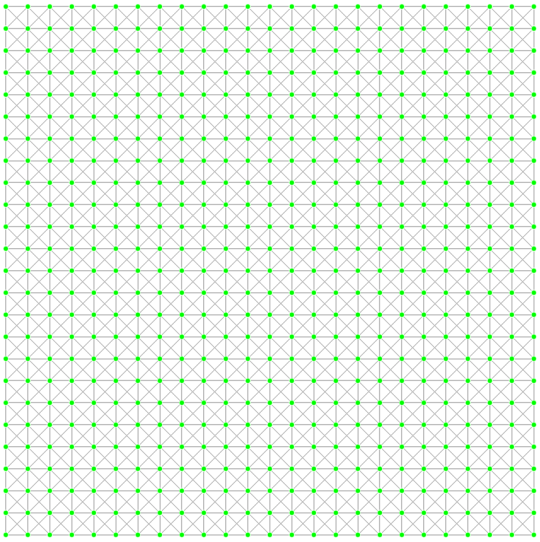


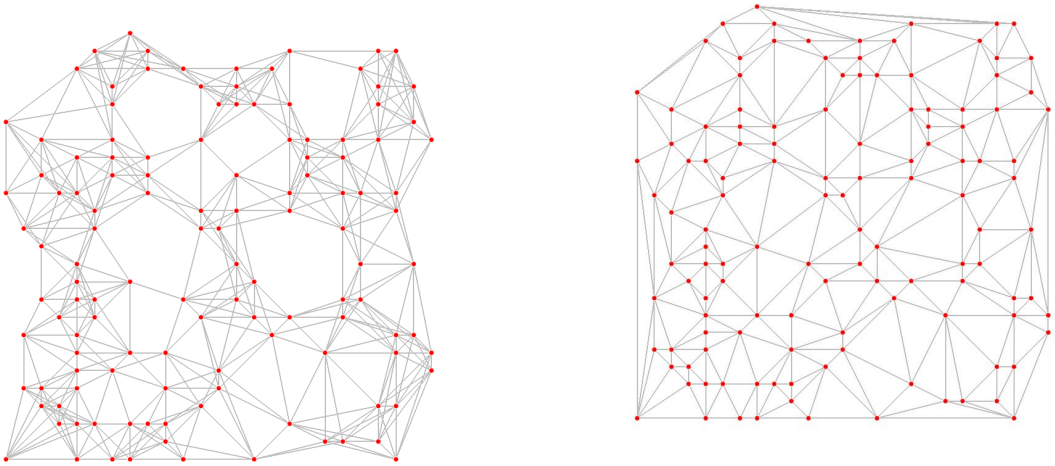
Figure 22. Illustration of edge construction in the image modality. Green points represent image pixels, and lines between points are edges.

The second modality is unstructured, which greatly complicates the graph creation process. The options considered were to construct a graph using a k-Nearest Neighbours (kNN) graph [69] or to create edges using Delaunay triangulation [70].

- **kNN graph** - for each node, edges are formed to k other, most similar nodes [69]. In this case, the similarity measure is Euclidean distance.
- **Delaunay triangulation** - while not exactly a graph construction method, it can be used as such if we consider graph nodes to be points in space. Delaunay triangulation is a method of creating a triangulation of a set of points, where no point is inside the circumcircle of any triangle. It maximizes the smallest angle in any of the triangles [70], which leads to a more uniform distribution of edges.

Examples of triangulation using both approaches can be seen in Figure 23. It was decided to use Delaunay triangulation, as it produces a much cleaner graph with edges spread all around most of the points, which allows for simpler usage of information from diverse neighbourhoods.

The last type of edge should create edges between nodes of different modalities, which are spatially close to each other. For the sake of conciseness, let the graph representing



(a) Graph constructed using 8-nearest neighbors.

(b) Graph constructed with Delaunay triangulation.

Figure 23. Different types of graph constructions.

the first (image) modality be denoted by M1, and the graph representing the second (unstructured) modality be M2. The edge creation process begins by placing both graphs into a one-higher dimensional space, where the added dimension coordinate was chosen to be 0 for M1 and 1 for M2. Next, an edge is created for every node of M1 with the closest node of M2. To ensure that no point is left without an edge, the same is done vice versa, and duplicates are later deleted. Moreover, edges are added an attribute (or weight), in the form of its Euclidean distance between connected nodes. Figure 24 illustrates the resulting graph.

Node attributes

In order to properly utilize all available information, the nodes need to contain the data they represent. Because of the possible non-commensurability of modalities, there is no straightforward approach for the representation of both modalities in a single graph. Two main possibilities were considered.

1. Heterogeneous graph - two types of nodes would exist, one for each modality
2. Zero padding - features from both modalities would be concatenated, filling zeroes to places where data from either modality is not available.

The second option was chosen, because of its relative simplicity and the fact that it allows for the usage of the same algorithms as in the case of the homogeneous graph. To illus-

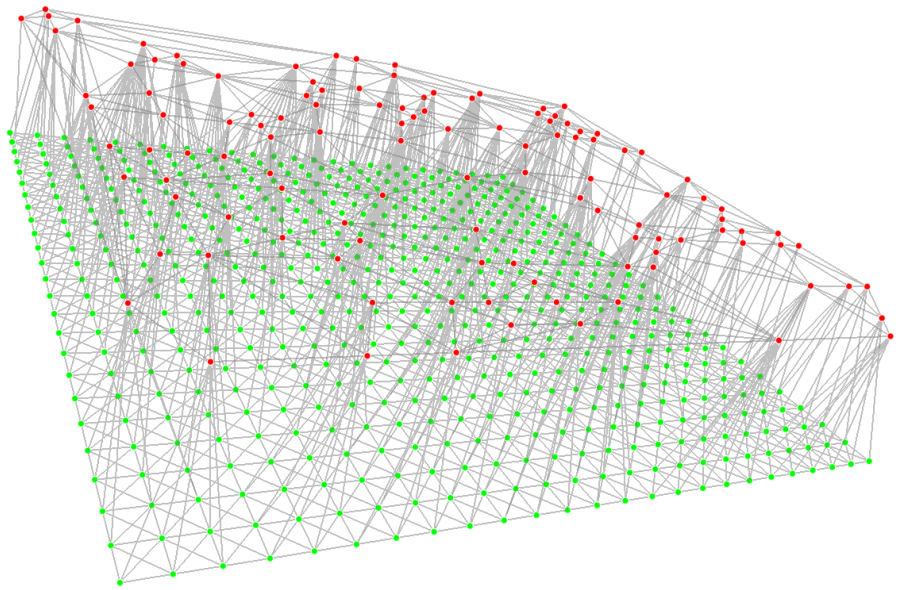


Figure 24. Illustration of a fully constructed graph with edges between both modalities. Green points represent the image, and red points the other modality.

trate, in the case of the RGB-D data, the image part of the graph nodes would contain a tuple $(r, g, b, 0)$, where r, g and b are the values of the respective image channels. The nodes of the graph representing the depth layer would contain tuple $(0, 0, 0, d)$, where d is the depth measurement.

5.4 Graph processing

The graph representation is fine, but useless in itself if it is not processed. In this work, a GNN was used to process the graph. A choice was made to utilize GAT, because of its ability to capture spatial relationships between nodes, as well as the expectation that the attention mechanism would be able to distinguish between important and less important graph edges. Even though in this thesis only GAT was used, the method is not restricted to this architecture, any graph processing technique can be used. The output of the processing can be a joint representation of modalities, an augmented version of a single modality, or even direct segmentation. The part of the graph representing an image is to be extracted after the processing, obtaining a grid-like representation of data or a result.

5.5 Framework summary

The proposed framework is designed to be as general as possible, with a modular structure that allows for customization and combination with various other methods. The framework is based on a graph representation of both modalities, with edges constructed to leverage spatial relationships between data nodes. The graph is then processed using a graph processing technique, such as a GAT, to obtain a fused representation of the modalities. The framework allows for both early and late fusion, as well as the incorporation of additional processing blocks for feature extraction or other purposes. The final output of the framework is in this case a grid-like representation of the data extracted from the graph, suitable for image segmentation.

6 Experiments

Multiple experiments were conducted to evaluate the proposed framework. Two datasets were used, each from a completely different field with independent use cases. The experiments were performed to find out how well the method works under various conditions, with data that have diverse characteristics. Experiments show the potential of the solution, producing excellent results on the mineralogy dataset, although the second experiment with timber data failed to deliver an improvement compared to the baseline solution. Nevertheless, the evaluation results demonstrate the capabilities of the method and show that the thesis objectives have been fulfilled.

In this chapter, the datasets are introduced, followed by the evaluation criteria and tools used. Lastly, the experiments and results are described and presented.

6.1 Data

The proposed solution was tested on two independent datasets. The first one is a mineralogy dataset, which contains Energy-Dispersive X-Ray Spectroscopy (EDS) data and greyscale image data from Backscattered Electrons (BSE). The second dataset is a timber dataset, which contains RGB and depth data.

Mineralogy dataset

The dataset contains a scan of a quartz sample from the Pitinga deposit. The original source of the dataset is [71]. The scan was performed with a Scanning Electron Microscope (SEM) Tescan TIMA. It contains various modalities, of which two were used: BSE measurements and EDS spectral measurements. The dataset contains 1596 measurements of the various non-intersecting locations on the sample. Each measurement has a 150x150 pixel big BSE image (one channel), with corresponding EDS spectrum for each pixel. The spectrum represents photon energies from 0 to 30 000 eV in 10 eV wide channels, so a single spectrum is an array of 3000 channels per pixel. An example of such a spectrum can be seen in Figure 25. Furthermore, the samples underwent liberation analysis in Tescan TIMA software, producing a mineral phase map of the sample. The map contains segmentation of the BSE image, segmenting the minerals in the sample, which was used as a ground truth for the segmentation task. In the whole dataset, there are 50 distinct

classes of minerals. Illustration of BSE image with corresponding segmentation can be seen in Figure 26.

One thing to note is that in a few cases, the measurements may be invalid. For example, when an edge of the sample is scanned, the spectral measurements from "beyond the edge" are not valid and may not contain any data at all. These areas were excluded from the evaluation.

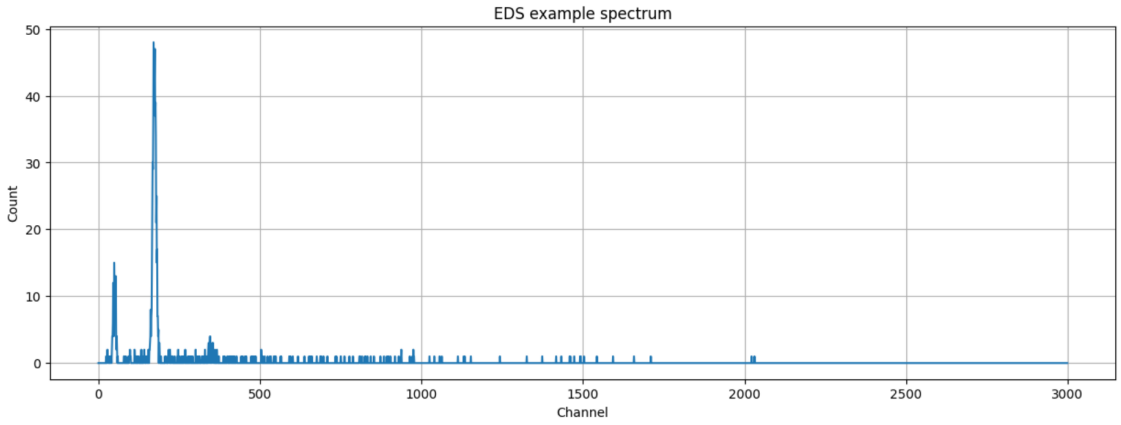
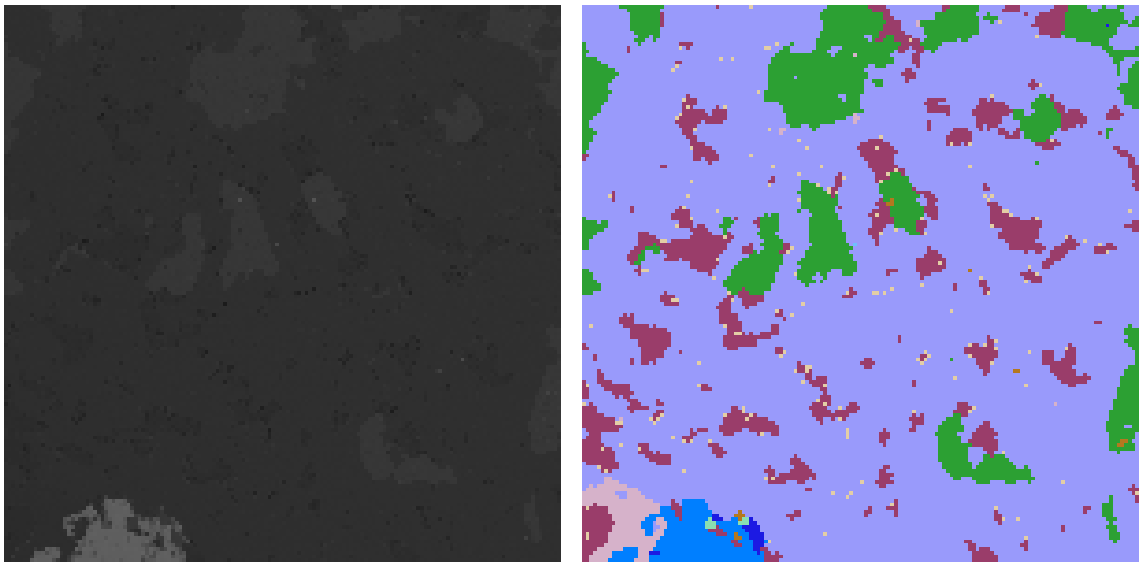


Figure 25. Example of a single pixel EDS spectrum.



(a) Example BSE image from the dataset. (b) Mineral segmentation for the image.

Figure 26. Example BSE image with the segmentation.

Modalities description BSE images are acquired quickly, but their big disadvantage is that they do not contain as precise information about the sample chemical composition

as EDS data. A BSE pixel value is dependent (among other things) on the mean atomic number $\overline{Z}$ of the scanned sample [72]. This creates a contrast in the image in places where the $\overline{Z}$ is different, which, however, is not the case for every different material. On the other hand, EDS data provide much more detailed information about the material composition, but the time requirements for acquiring this modality are much bigger. For this reason, the acquired EDS data are often sparser than the BSE data. Data fusion of these modalities can provide a more complete picture of the sample with sparser data, as both modalities can potentially benefit from each other.

Timber dataset

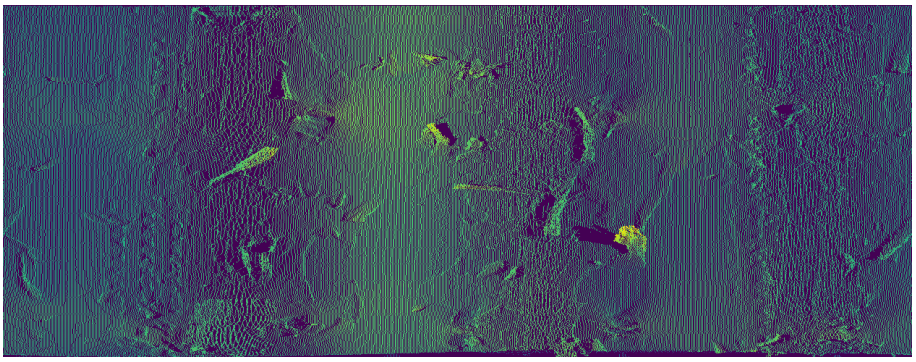
This dataset contains seven scans of sawn timber made from Scots Pine trees. The modalities present are the RGB images of the timber, as well as sparse heightmaps and masks, which mark the locations of knots on the timber. Modalities were acquired by scanning the timber, which was rotated around its lengthwise axis. The data were then stitched together, creating the resulting RGB image and a pointcloud. The pointcloud was then projected on the image, creating a sparse heightmap (approximately 9 times sparser than the image). The ground truth (knot mask), was acquired from X-Ray Computed Tomography (CT) scan, acquired in the same laboratory setup. The output of the CT scan was used for internal timber structure reconstruction, from which in turn the knot mask was generated. The modalities were aligned prior to their usage in any model.

An example from the dataset can be seen in Figure 27. The width of images is 1256px and the height is between 487 and 564px, depending on the timber height.

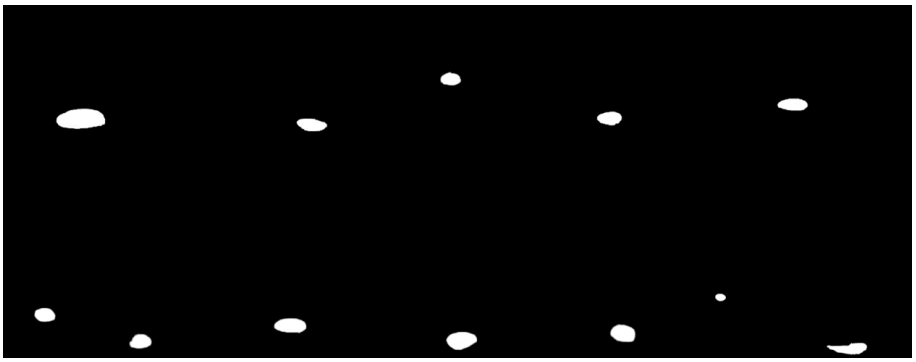
6.2 Evaluation criteria

The metrics were selected to quantify the performance of models in concrete tasks, in order to create a fair comparison with reference models. The evaluation was done using standard metrics for segmentation tasks. The metrics used, depending on the experiment, were Recall, Precision, F-1 Score, Intersection over Union (IoU) and Receiver Operating Characteristic (ROC) curve. All of the mentioned metrics are based on items from the Confusion Matrix. A confusion matrix is a table that is often used to describe the performance of a classification model on a set of data for which the true values are known. The confusion matrix consists of four parts: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) samples. The definitions of these terms are as

(a) RGB image



(b) Sparse heightmap



(c) Mask

Figure 27. Modalities for log number 1. The scan represents vertically placed timber, meaning that the y-axis represents the height and the x-axis the place on the circumference.

follows: TP is the number of correctly predicted positive samples, TN is the number of correctly predicted negative samples, FP is the number of incorrectly predicted positive samples and FN is the number of incorrectly predicted negative samples. The confusion matrix is shown in Table 1. Note, that while the illustrated confusion matrix is for binary classification tasks, but similar approach can be applied to multiclass classification. A metric needs to be calculated for each class independently, effectively treating the problem as a binary classification problem for each class. Optionally, the results can be then reduced to a single value, for example using average or weighted average.

Table 1. Confusion Matrix for binary classification.

		Predicted	
		Positive	**Negative**
Actual	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Recall

Recall (also known as Sensitivity or True Positive Rate) measures the ability of the model to capture the positive instances. It is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

Precision

Precision (also known as Positive Predictive Value) measures the relevancy of positively classified instances. It is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

F-1 Score

The F-1 Score is the harmonic mean of Recall and Precision. It is defined as:

$$\text{F-1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

IoU

IoU (also known as the Jaccard Index) is a measure of the overlap between the predicted and ground truth masks. It is defined as:

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{11}$$

ROC

ROC curve is a graphical representation of the performance of a binary classification model. It is created by plotting the True Positive Rate (Recall) against the False Positive Rate for different threshold values. The area under the ROC curve is a measure of the model's performance. The area under the curve is 1 for a perfect model and 0.5 for a random model.

The false positive rate is calculated as:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{12}$$

Connected to the ROC is Area Under the Receiver Operating Characteristic (AUROC), which is another measure of the model's performance. A value of 1 is typical for a perfect model while 0.5 is typical for a random model.

6.3 Implementation details

Implementations of the framework were done in Python 3.9, with PyTorch [73] and PyTorch Geometric [74]. For training a library PyTorch Lightning was utilized [75]. The mineralogy dataset loading was done with a modified code published with [1]. Graph construction was done with the help of SciPy library [76]. Evaluation metrics were calculated with the TorchMetrics library [77].

6.4 Experiment: EDS + RGB dataset

The first experiment was done on the mineralogy dataset. Since the dataset contains spectral measurements that are, in fact, structured, it had to be preprocessed to simulate unstructured modality. First, a parameter was chosen that described a fraction of the spectral data that should be utilized. According to the parameter, random spectral data points were chosen for further processing. This simulates the process where a small set of randomly selected spectral measurements are made to speed up the measurement. Because of the randomness, the structured nature of the data was effectively removed. Furthermore, this process was done for each invocation of the getter method, leading to a new random spectral data selection. This allowed for better utilization of the provided data and worked as a measure against overfitting of the model.

The dataset was split into training, validation and testing sets, with 80% of the data used for training, 10% for validation and 10% for testing.

Framework implementation

For this framework implementation, one optional processing step was used. The EDS modality is quite large (3000 channels), especially in comparison to the BSE modality. This would lead to a higher computational and memory cost for training as well as for inference. Because of this, an embedding network by Juranek *et al*. [1] was employed for dimensionality reduction. The network reduces dimensionality from the original 3000 channels to 64 channels while preserving the most important information. The dataset implementation illustration can be seen in Figure 28.

Graph construction was done as previously described. As a graph processing method, a

GAT was utilized. The network contained 3 layers, each with 16 hidden channels and 2 attention heads. Moreover, while processing the graph, self-loop edges were added with weight 0 in order to allow the node to pass messages to itself with the highest priority. The last layer output was a 50-channel vector for each pixel, each presenting a single mineral class from the dataset. While training, the BSE modality was left complete and without change, while the EDS modality was decimated to contain anywhere between 0% to 70% of original data.

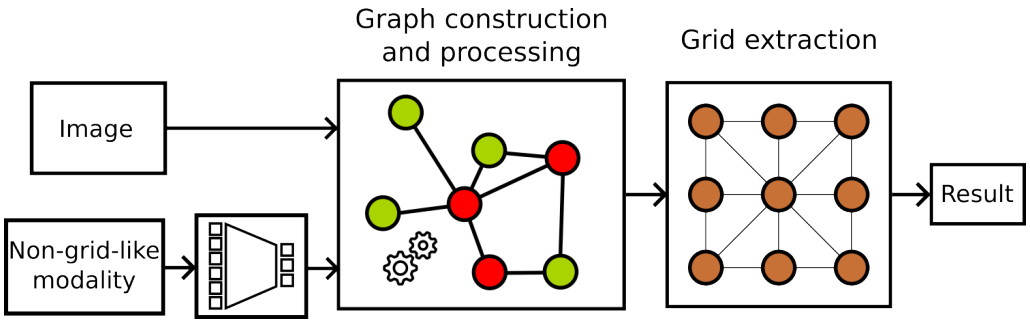


Figure 28. Implementation of the framework for mineralogy dataset. One optional part was used for dimensionality reduction of the EDS modality.

Results

The trained model exhibits clear signs of successful data fusion. Example outputs can be seen in Figure 29. The Figure shows the input BSE image, as well as the ground truth for the modality pair. The rest are generated segmentations with different percentages of EDS modality used. It can be seen, that segmentation produced with just 0.1% of EDS data is extremely rough, but the results are quickly improving with added data. The results are shown in Figures 31, 32 and 33. It is shown, that the model is able to learn from the EDS data, as the performance of the model increases with the percentage of EDS data. With only 1% of EDS data, the model can achieve quite high rankings in all evaluation metrics. With added data, the performance increases quite sharply in a logarithmic fashion, which agrees with the expectations. Note that the X-axis is not linear.

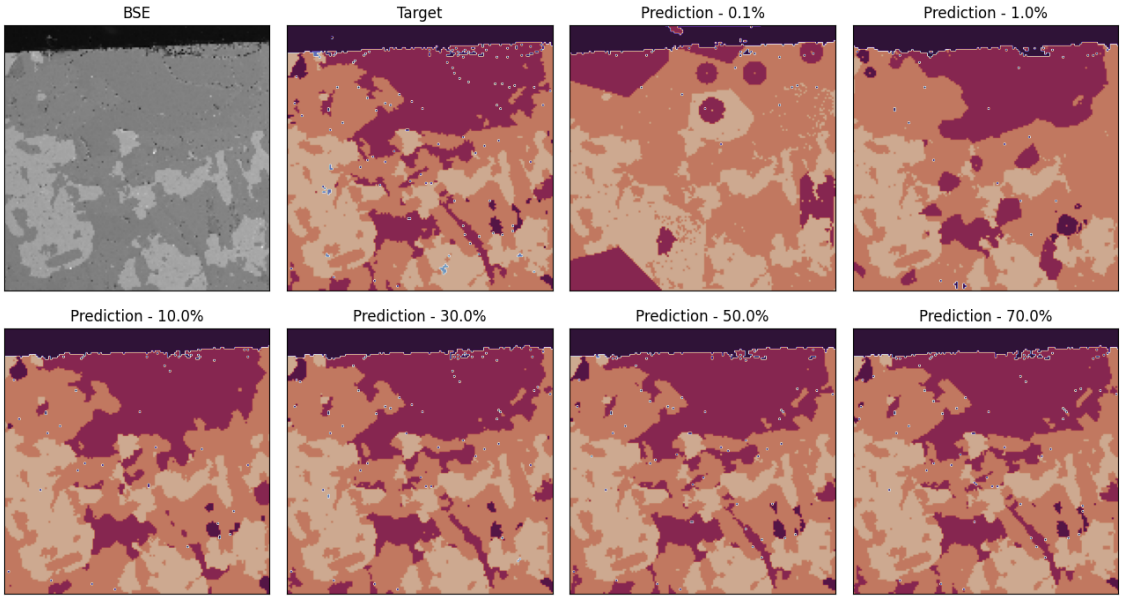


Figure 29. Visualization of outputs of the framework. The top left is BSE image, to the right of it is the ground truth, and other images are outputs with various percentages of input EDS data.

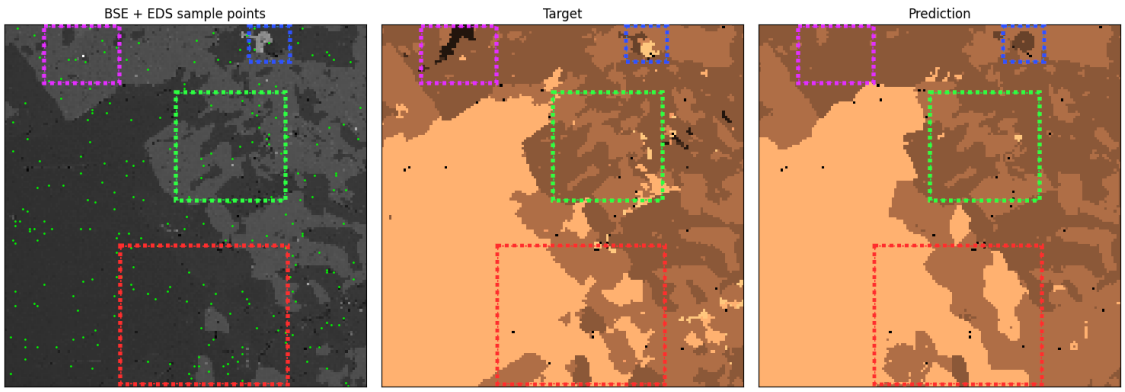


Figure 30. A singular example of a prediction made with 1% of the EDS data. Green points on the image show locations of EDS measurements. The magenta frame highlights a class that was entirely missed by the prediction. The green one shows the area where the prediction is remarkably precise. The blue frame shows an interesting example, where EDS data barely missed one class and prediction was to able to infer it correctly. The red highlight marks an area that contains two classes nearly indistinguishable on the BSE image and as a result, the border between classes is quite smooth.

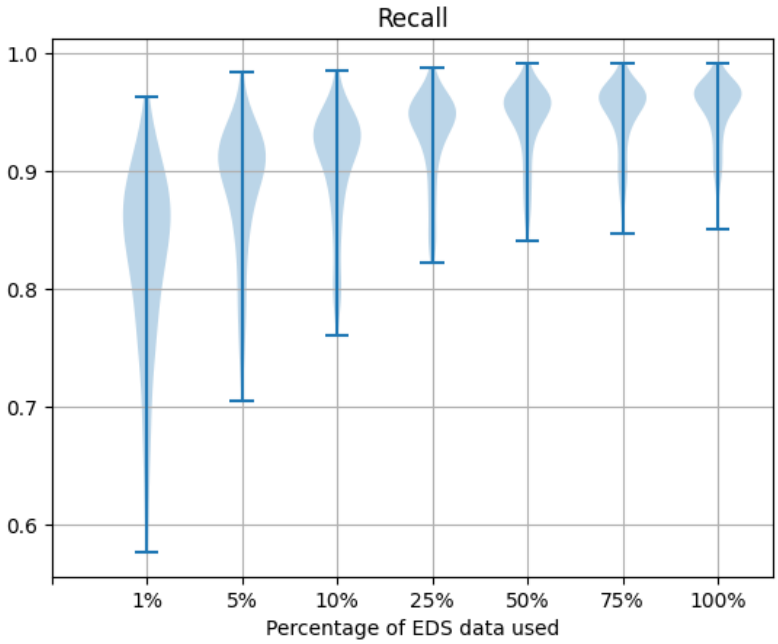


Figure 31. Recall of the implemented framework with different EDS percentages.

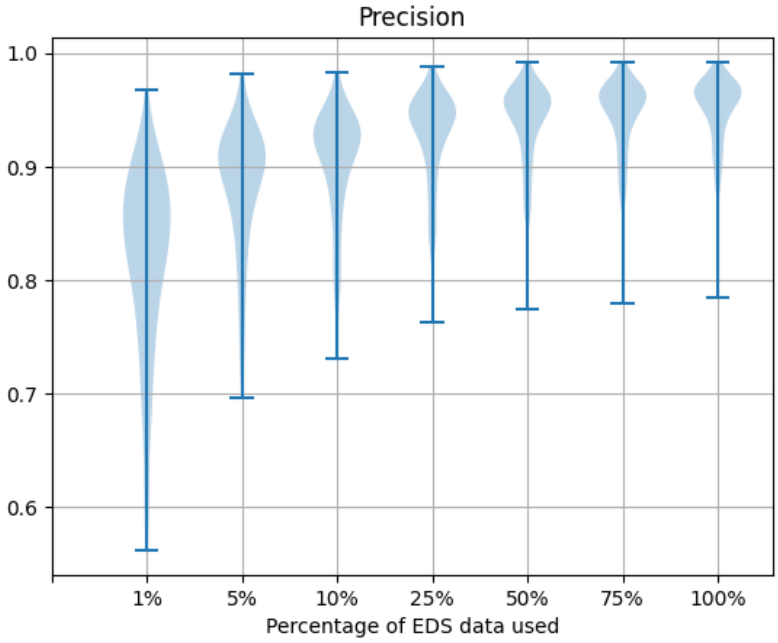


Figure 32. Precision of the implemented framework with different EDS percentages.

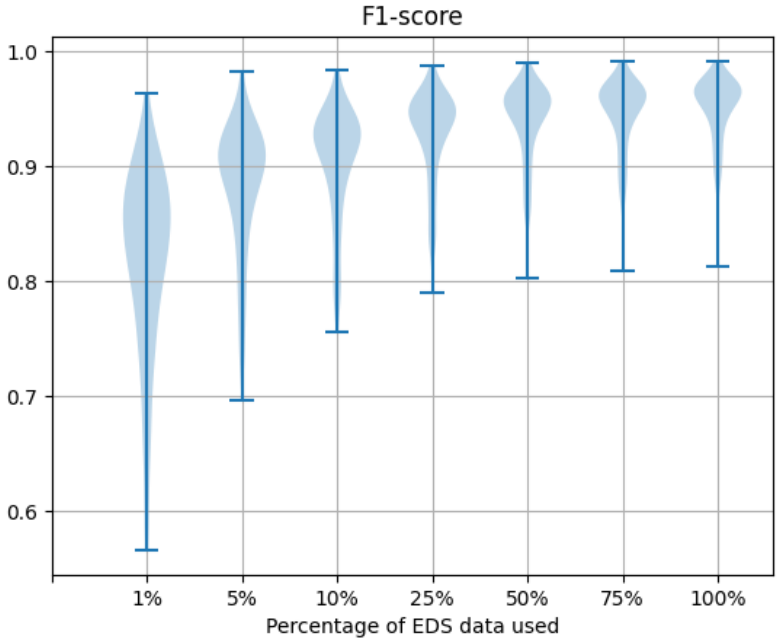


Figure 33. F-1 score of the implemented framework with different EDS percentages.

Comparison with GDLS

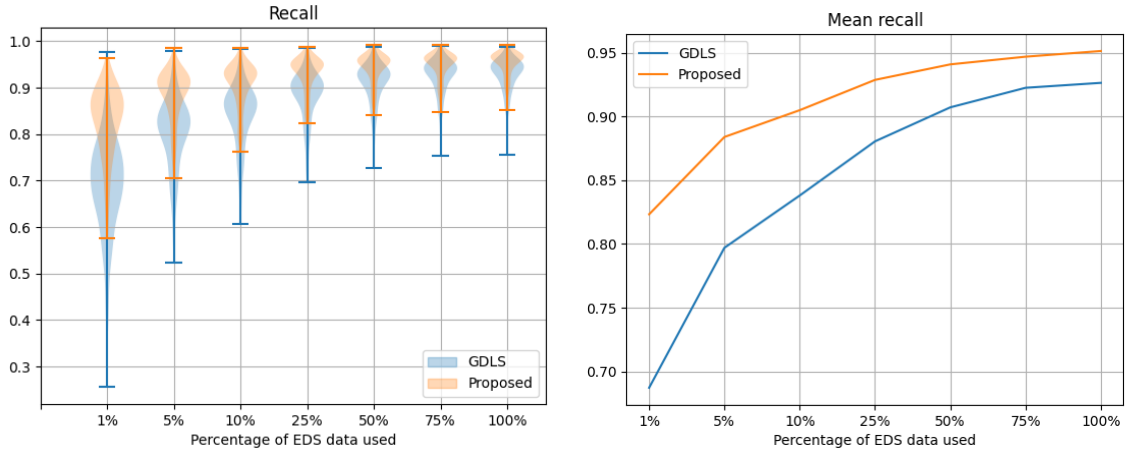
Moreover, a comparison with the Graph-based Deep Learning Segmentation (GDLS) method described in [1] was done. The GDLS works similarly in a few ways but is based around Markov Random Field segmentation. EDS data are first embedded into a lower dimensional space using a CNN (which was used in this solution as well), then a graph is constructed from the EDS data locations and data relevant to these locations. The graph is then processed, disconnecting edges with substantial dissimilarities. The results of this processing are labelled segments which mark components made out of the same material. Finally, the Markov Random Field is run on a fully connected grid of BSE measurements initialized with labels from the previous section. The result is a segmented BSE image. The segments denote parts of the image with the same material, however, it does not tell what material. To classify these segments into mineral phases, additional processing is required.

It is important to note that various assumptions had to be made in order to make methods comparable. First of all, as mentioned previously, GDLS produces segments without any semantic meaning, whereas the proposed method provides direct segmentation to mineral phases. To alleviate this problem, the output of the GDLS was sent to the Tescan company, where the segments were transformed to the representation matching output of the proposed method. This process required second access to the full EDS data. Second of all,

the GDLS has two parameters influencing the final segmentation. The parameters were selected on a few samples and used for the whole dataset in such a way, which prioritized a higher number of segments. It is possible, that other parameters would create better segmentation results for singular samples, but selected parameters should be enough to provide a good comparison. Lastly, the proposed method is not nearly as general as GDLS, as it was trained on a dataset with a limited number of classes.

Comparison results

Evaluation with recall metric shows that the proposed method significantly outperforms the GDLS in all cases, as shown in Figure 34.



(a) Recall comparison.

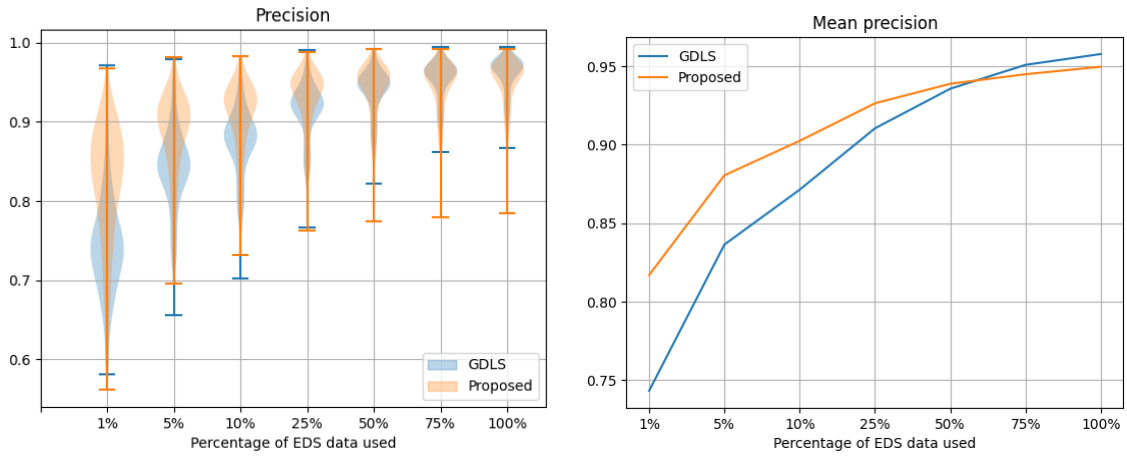
(b) Mean recall comparison.

Figure 34. Comparison of recalls of the implemented framework with GDLS. In this metric, the proposed method significantly outperforms the GDLS in all cases.

However, with precision, the result is not as definite. With a lower number of EDS data points, the proposed method produces better results, however with an increasing number of data points situation changes. In the highest counts, the GDLS performance is better than the proposed method, as shown in Figure 35.

The F1 score shows that the GDLS method performs better than the proposed method, as shown in Figure 36.

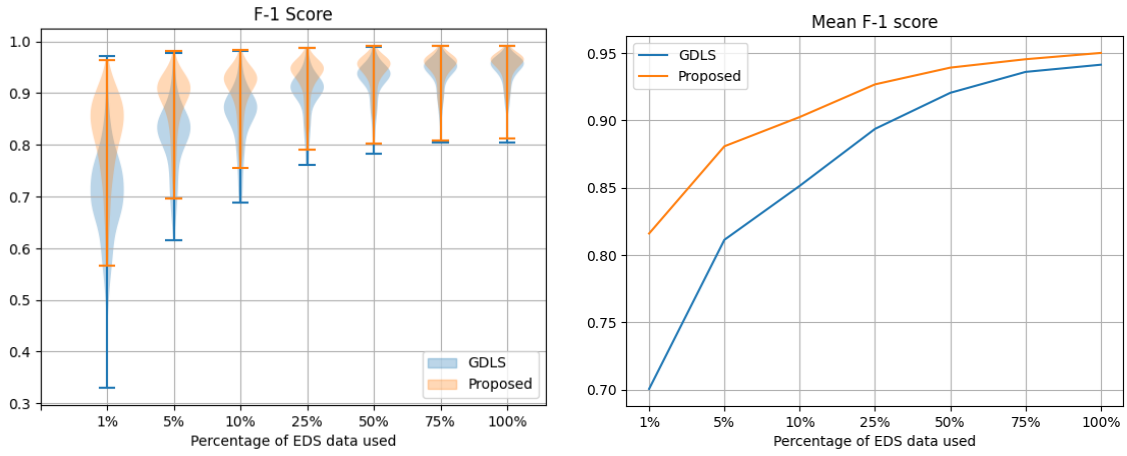
Example outputs of both methods side by side can be seen in Figure 37.



(a) Precision comparison.

(b) Mean precision comparison.

Figure 35. Precision of the implemented framework compared with GDLS. With a lower number of EDS data points, the proposed method produces better results, however with an increasing number of data points situation changes. In the highest counts, the GDLS performs better than the proposed method.



(a) F1 comparison.

(b) Mean f1 comparison.

Figure 36. F-1 score of the implemented framework with different EDS percentages.

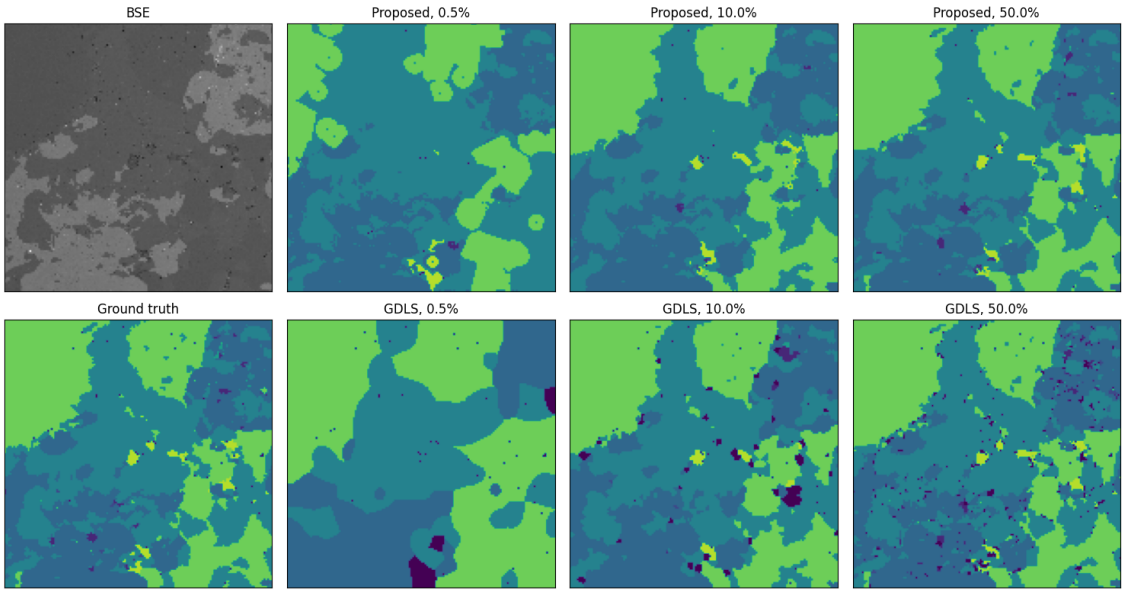


Figure 37. Example outputs of the proposed method with GDLS, for various EDS data percentages. The leftmost column shows the input BSE image and the ground truth. Otherwise, the top row shows the output of the proposed method, and the bottom row shows the output of the GDLS.

6.5 Experiment: Timber dataset

The second experiment was conducted on the timber dataset. The task is to predict the locations of knots, based on the RGB image and sparse heightmap.

The dataset itself posed numerous challenges, the most obvious one being the dataset size. The dataset contains only seven samples, which may cause trouble for the generalization of trained solutions, as models have a very high tendency to overfit on such a small dataset. This was mitigated by using small patches of the sample as training data, and utilization of data augmentation. The patch generation algorithm first rotated the whole sample by a random amount, after which a patch of a given size was extracted. It is important to note that while the graph representation would theoretically allow for a "native" representation of the log image, that is the borders of the image graph could be joined to simulate wrapping, the patchwise processing meant that this was not possible.

The second challenge of the dataset was the class imbalance. The knots are small and relatively far from each other, which led to the not-knot class being much more prevalent than the knot class, almost by a factor of 140. This was mitigated by assigning a weight to each class when calculating the loss function (Binary Cross Entropy), namely the positive class had a weight of 140 and the negative class weight was 1.

The dataset was split into training, testing and validation partitions as well. The first 5 logs were designated for training, the sixth for validation and the final one for testing.

Framework implementation

Numerous implementations were tried. The first implementation of the framework was, similar to the previously described experiment, just the GNN, without any optional processing steps (illustrated in Figure 38). This version failed to learn on the dataset, which may be caused by the fact that the GNN is not designed to handle image data, and of course by the aforementioned challenges of the dataset.

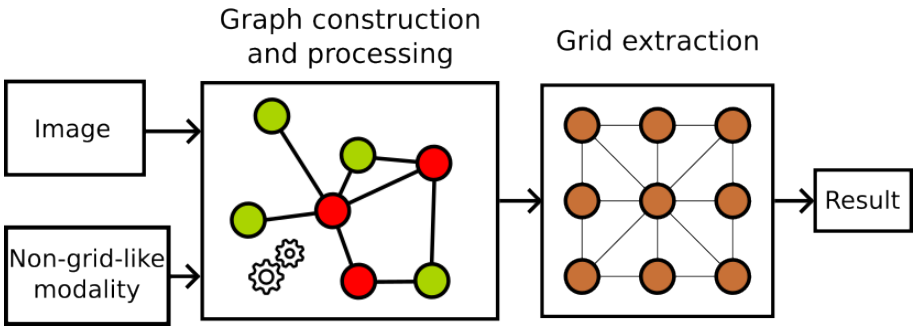


Figure 38. First implementation of the framework. All the optional parts were omitted.

Further investigation revealed that it is very difficult to create *any* model, which would converge to an acceptable result. From the few working models, the best performing turned out to be the U-Net, created from the VGG-16 model pre-trained on the ImageNet [78].

Because of this, the second version of the framework implementation tried to use this U-Net as an *optional processing step*. The idea was to take the output of the U-Net and use its features to ease the load on the fusion step, as illustrated in Figure 39.

The second option where to integrate the U-Net model into the framework was the last optional processing step, which processes the joint representation of both modalities. This led to the third version of the framework implementation, which is illustrated in Figure 40. The broad idea was then to use the GNN to "enhance" the image data, which could lead to a better overall performance.

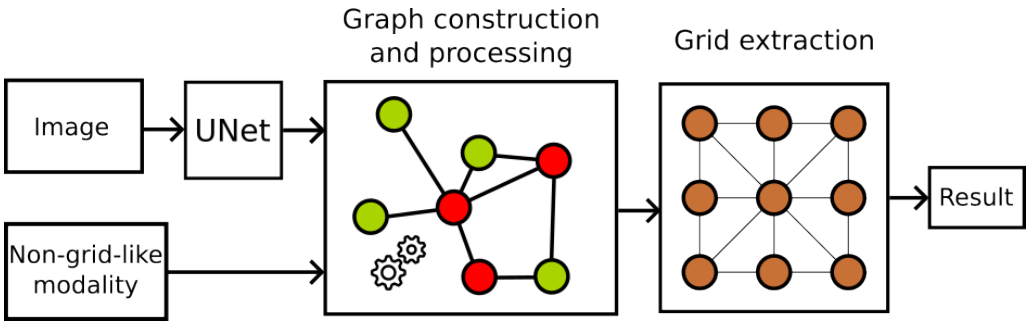


Figure 39. Second implementation of the framework.

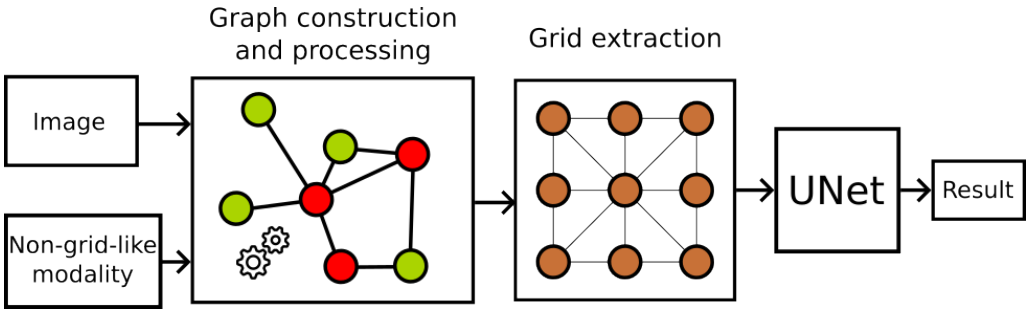


Figure 40. Third implementation of the framework.

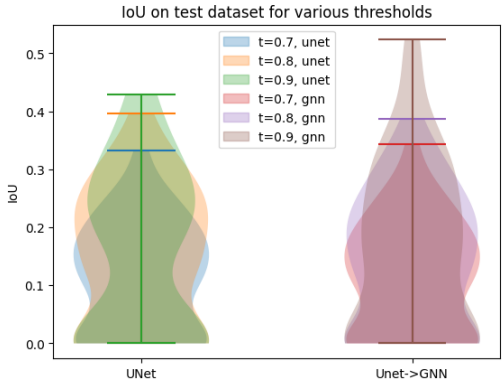
Results

As mentioned previously, the first framework implementation without any optional processing step failed to deliver any meaningful result, so for this reason, the evaluation of this model will be omitted.

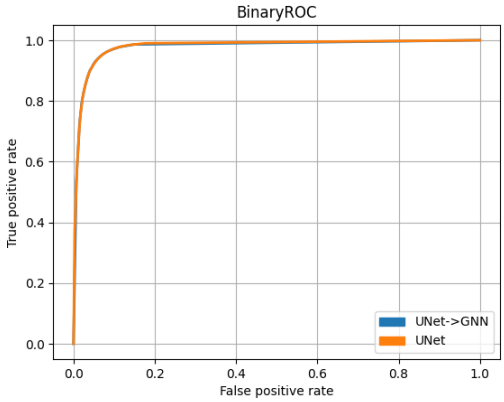
The evaluation for the second implementation was done by plotting the distribution of IoU metric over the testing set. While this metric proved that the models were different in some way, it was unclear whether there was any improvement. Further investigation using ROC plot proved that the performance of the models is almost identical. Both metrics are visualized in Figure 41.

The third implementation was evaluated similarly, producing very much the same results. Metrics are shown in Figure 42.

In the end, the best result the framework was able to achieve was to make the GAT step transparent. Example outputs of the U-Net and implemented frameworks can be seen in Figure 43.

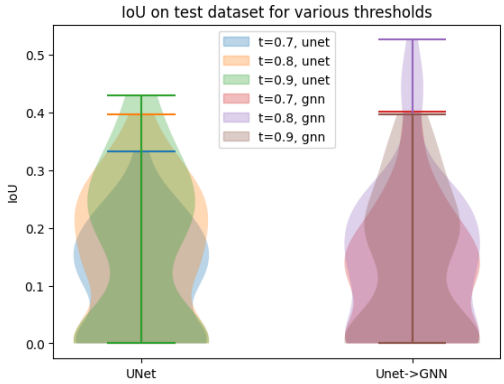


(a) IoU distribution compared with U-Net model.

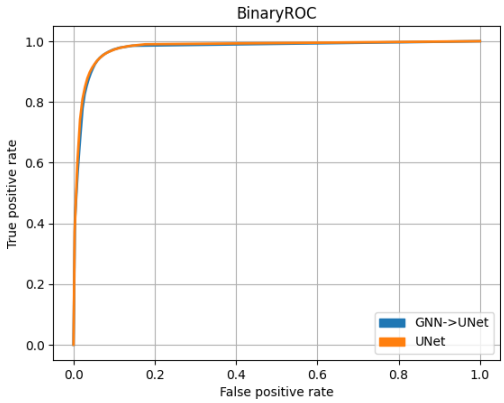


(b) ROC plot of the U-Net and fusion models.

Figure 41. Evaluation of the second framework implementation.

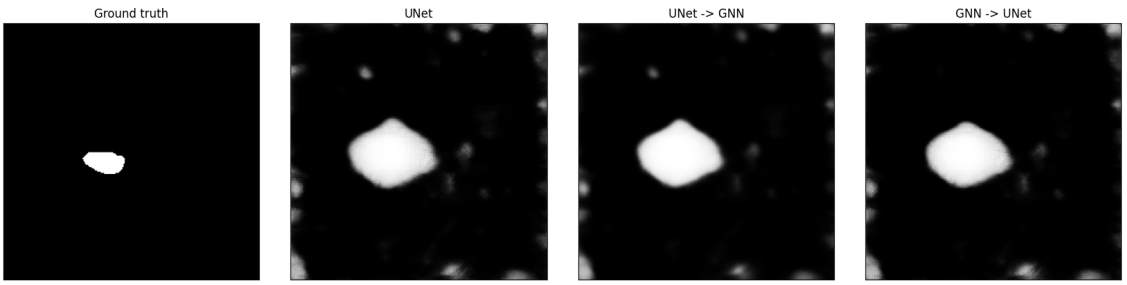


(a) IoU distribution compared with U-Net model.



(b) ROC plot of the U-Net and fusion models.

Figure 42. Evaluation of the second framework implementation.



(a) Ground truth.

(b) UNet output.

(c) Output of the framework with UNet before graph stage.

(d) Output of the framework with UNet after graph stage.

Figure 43. Example outputs of the baseline UNet and implemented framework variants.

7 Discussion

This chapter reviews and analyses the results obtained in this work. Various aspects of the work are considered, together with hypotheses about the shortcomings of the solution. Finally, potential future contributions are explored.

7.1 Current study

Multimodal data fusion has an incredible potential to improve the performance of various machine learning tasks, including image segmentation. A general method of fusing various data modalities could help in many fields in various ways, be it in accuracy, time requirements, or others.

In this thesis, a general framework for the fusion of image and non-grid-like data was proposed. The framework was designed to be as general as possible, with a modular structure that allows for customization and combination with various other methods. The core of the framework is based on a graph constructed to express the inter and intra-modality relationships as accurately as possible. The graph is subsequently processed with a GNN, aiming to mix the information from both modalities into a joint representation, or a direct segmentation. The grid-like representation is acquired by extracting the part of the graph originally representing the image modality.

The approach has been tested on two datasets, the first one being from the mineralogy domain and the second one from the timber-processing domain. In the first case, the results are encouraging, showing that the approach is capable of adapting and fusing the information from both modalities with excellent performance. The prediction quality of the implemented framework steadily increased with the density of provided EDS data, which aligns with the expectations. It also shows the speed/accuracy tradeoff, as less data can be acquired quicker, but the results will be less precise. The approach was also compared with GDLS method described in [1], showing that the proposed method outperforms the other method in almost all cases, except in precision metric with EDS density above 50%. Note, that various assumptions had to be made in order to make the solutions comparable, namely estimation of optimal parameters for the GDLS and the necessity for additional external steps to match the output formats of the method.

However, it is not a universal solution, as demonstrated by the experiment with the timber

dataset. Even though it is unclear if the failure to improve was caused by the approach itself, or the challenging nature of the data, the proposed method failed to deliver any improvement when compared with baseline U-Net. The most notable dataset challenges were its small size and a huge imbalance in dataset classes.

7.2 Future work

Continuation of this work should include more rigorous testing on other datasets, in order to better explore the flexibility, adaptability, and applicability of the proposed method in other fields.

While in this thesis a GAT was used, because of its simplicity and good performance, it is not the only option. The exploration of other GNN models or non-neural approaches could bring new insights into the problem, and potentially improve the results. Joint processing of nodes and edges could be beneficial, for example utilizing a Graph Network [79].

While in this work a zero-filling was utilized as a method of representation of both modalities in a single homogeneous graph, another approach could be a heterogeneous graph. This would mean an even more natural representation of the data, however, the processing would need to adapt.

The proposed method explored image segmentation as a vertex classification task. A more general approach would be taking the task as edge classification. The output of the graph processing would be a graph with edges assigned the probability of connected vertices belonging to the same segments. This would be a better approach for segmentation with an unknown number of classes, which is the case for example in the mineralogy dataset.

The solution is also not limited to only two modalities. The extension to more modalities is straightforward, the only change necessary would be a multi-step inter-modality edge construction, which would have to be done for every combination of two modalities.

8 Conclusion

The goal of this thesis was to create a flexible framework for multimodal data fusion, with one modality being an image and the second one a general, unstructured modality. As such, the goal was fulfilled.

To sum up, in the first part of the thesis, the state-of-the-art concerning the topic of this work was reviewed, including multimodal data, neural networks, data segmentation, and data fusion. The second part of the work introduces a novel framework for the fusion of images and other non-grid-like modalities. At its core, it uses graphs for data representation and relationship modelling, and GNNs for graph processing. Moreover, it contains optional blocks for additional feature extraction or dimensionality reduction, which can be used to improve the results. Subsequently, it was evaluated on two independent datasets, producing excellent results in one case and insufficient in the second case, showing the potential of the solution but also its limitations.

The presented joint graph representation is very flexible, allowing for a wide variety of structures, thus it is also suitable for modalities with missing values. The data processing method used is very flexible, which makes it possible to use non-commensurable modalities as well. With this in mind, the goals set at the beginning of framework creation have been achieved. The framework is general and modular, allowing for customization and combination with various other methods.

Possible continuation of this work may lie for example in exploring other GNN architectures, testing it on other multimodal datasets, or investigating heterogenous graph applications. Moreover, edge classification can be used instead of node classification, or the framework can be adapted to more than two modalities.

REFERENCES

[1] Roman Juránek, Jakub Výravský, Martin Kolář, David Motl, and Pavel Zemčík. Graph-based deep learning segmentation of EDS spectral images for automated mineral phase analysis. *Computers & Geosciences*, 165:105109, 2022.

[2] Yuxiang Sun, Weixun Zuo, Peng Yun, Hengli Wang, and Ming Liu. FuseSeg: Semantic Segmentation of Urban Scenes Based on RGB and Thermal Data Fusion. *IEEE Transactions on Automation Science and Engineering*, 18(3):1000–1011, 2021.

[3] Jinming Cao, Hanchao Leng, Daniel Cohen-Or, Dani Lischinski, Ying Chen, Changhe Tu, and Yangyan Li. RGB×D: Learning depth-weighted RGB patches for RGB-D indoor semantic segmentation. *Neurocomputing*, 462:568–580, 2021.

[4] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and Dense Data with CNNs: Depth Completion and Semantic Segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018.

[5] Yuxiang Sun, Weixun Zuo, and Ming Liu. RTFNet: RGB-Thermal Fusion Network for Semantic Segmentation of Urban Scenes. *IEEE Robotics and Automation Letters*, 4(3):2576–2583, 2019.

[6] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[7] Jean-Denis Durou, Maurizio Falcone, and Manuela Sagona. Numerical methods for shape-from-shading: A new survey with benchmarks. *Computer Vision and Image Understanding*, 109(1):22–43, 2008.

[8] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust Reconstruction of Indoor Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[9] Cristina Palmero, Albert Clapés, Chris Bahnsen, Andreas Møgelmose, Thomas B Moeslund, and Sergio Escalera. Multi-modal RGB–Depth–Thermal Human Body Segmentation. *International Journal of Computer Vision*, 118(2):217–239, June 2016.

[10] Ilan Shomorony, Elizabeth T. Cirulli, Lei Huang, Lori A. Napier, Robyn R. Heister, Michael Hicks, Isaac V. Cohen, Hung-Chun Yu, Christine Leon Swisher, Natalie M.

Schenker-Ahmed, Weizhong Li, Karen E. Nelson, Pamila Brar, Andrew M. Kahn, Timothy D. Spector, C. Thomas Caskey, J. Craig Venter, David S. Karow, Ewen F. Kirkness, and Naisha Shah. An unsupervised learning approach to identify novel signatures of health and disease from multimodal data. *Genome medicine*, 12(1):7–7, 2020.

[11] Léonard Boussioux, Cynthia Zeng, Théo Guénais, and Dimitris Bertsimas. Hurricane Forecasting: A Novel Multimodal Machine Learning Framework. *Weather and forecasting*, 37(6):817–831, 2022.

[12] Yan Zhang, Steffen Müller, Benedict Stephan, Horst-Michael Gross, and Gunther Notni. Point cloud hand–object segmentation using multimodal imaging with thermal and color data for safe robotic object handover. *Sensors*, 21(16):5676, 2021.

[13] Dana Lahat, Tülay Adalý, and Christian Jutten. Challenges in multimodal data fusion. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 101–105. IEEE, 2014.

[14] Don Speray and Steve Kennon. Volume probes: interactive data exploration on arbitrary grids. *SIGGRAPH Computer Graphics*, 24(5):5–12, nov 1990.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[16] H. Taud and J.F. Mas. Multilayer Perceptron (MLP). In María Teresa Camacho Olmedo, Martin Paegelow, Jean-François Mas, and Francisco Escobar, editors, *Geomatic Approaches for Modeling Land Change Scenarios*, pages 451–455. Springer International Publishing, 2018.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 25. Curran Associates, Inc., 2012.

[18] Isha Garg, Priyadarshini Panda, and Kaushik Roy. A Low Effort Approach to Structured CNN Design Using PCA. *IEEE Access*, 8:1347–1360, 2020.

[19] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR*. Cornell University, 2015.

[20] Max Ferguson, Ronay Ak, Yung-Tsun Tina Lee, and Kincho H. Law. Automatic localization of casting defects with convolutional neural networks. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1726–1735. IEEE, 2017.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.

[22] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587. IEEE, 2014.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. IEEE, jun 2016.

[24] Chien-Yao Wang and Hong-Yuan Mark Liao. YOLOv9: Learning what you want to learn using programmable gradient information. *arXiv:2402.13616*, 2024.

[25] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative Study of CNN and RNN for Natural Language Processing. *arXiv:1702.01923*, 2017.

[26] G. E. Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[27] Jing Gao, Peng Li, Zhikui Chen, and Jianing Zhang. A Survey on Deep Learning for Multimodal Data Fusion. *Neural Computation*, 32(5):829–864, 05 2020.

[28] G. E. Hinton. Boltzmann machine. *Scholarpedia*, 2(5):1668, 2007.

[29] Nan Zhang, Shifei Ding, Jian Zhang, and Yu Xue. An overview on Restricted Boltzmann Machines. *Neurocomputing*, 275:1186–1199, 2018.

[30] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep Learning for 3D Point Clouds: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2021.

[31] Zhizhong Kang, Juntao Yang, Ruofei Zhong, Yongxing Wu, Zhenwei Shi, and Roderik Lindenbergh. Voxel-Based Extraction and Classification of 3-D Pole-Like Objects From Mobile LiDAR Point Cloud Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(11):4287–4298, 2018.

[32] Yongming Rao, Jiwen Lu, and Jie Zhou. Spherical Fractal Convolutional Neural Networks for Point Cloud Recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 452–460. IEEE, 2019.

[33] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85. IEEE, 2017.

[34] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: deep hierarchical feature learning on point sets in a metric space. In *31st International Conference on Neural Information Processing Systems (NeurIPS)*, page 5105–5114. ACM, 2017.

[35] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling Point Clouds With Self-Attention and Gumbel Subset Sampling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3318–3327. IEEE, 2019.

[36] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5560–5568. IEEE, 2019.

[37] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep Parametric Continuous Convolutional Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2589–2597. IEEE, 2018.

[38] Alexandre Boulch. ConvPoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 88:24–34, 2020.

[39] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-Shape Convolutional Neural Network for Point Cloud Analysis. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8887–8896. IEEE, 2019.

[40] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A Gentle Introduction to Graph Neural Networks. *Distill*, 2021. https://distill.pub/2021/gnn-intro, accessed February, 13, 2024.

[41] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[42] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Cornell University, 2015.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 30. Curran Associates, Inc., 2017.

[44] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *arXiv:1710.10903*, 2018.

[45] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

[46] Vladimir Gligorijević, P. Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C. Taylor, Ian M. Fisk, Hera Vlamakis, Ramnik J. Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, 12(1):3168, May 2021.

[47] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In *37th International Conference on Machine Learning*, ICML'20. ACM, 2020.

[48] Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. Pixie: A System for Recommending 3+ Billion Items to 200+ Million Users in Real-Time. In *2018 World Wide Web Conference (WWW '18)*, page 1775–1784. ACM, 2018.

[49] Benjamin Graham and Laurens van der Maaten. Submanifold Sparse Convolutional Networks. *arXiv:1706.01307*, 2017.

[50] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Penksy. Sparse Convolutional Neural Networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 806–814. IEEE, 2015.

[51] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image Segmentation Using Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022.

[52] Amanda Boyd. Seals at Nantucket National Wildlife Refuge. https://www.flickr.com/photos/43322816@N08/5961318915, 2024. [Online; accessed February, 13, 2024].

[53] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[54] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE, 2017.

[55] Manuel Diaz-Zapata, Özgür Erkent, and Christian Laugier. Yolo-based panoptic segmentation network. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1230–1234. IEEE, 2021.

[56] H. B. (Harvey B.) Mitchell. *Multi-sensor data fusion : an introduction*. Springer, 2007.

[57] Junwei Duan, Shuqi Mao, Junwei Jin, Zhiguo Zhou, Long Chen, and C. L. Philip Chen. A Novel GA-Based Optimized Approach for Regional Multimodal Medical Image Fusion With Superpixel Segmentation. *IEEE Access*, 9:96353–96366, 2021.

[58] Geoffrey Iyer, Jocelyn Chanussot, and Andrea L. Bertozzi. A Graph-Based Approach for Data Fusion and Segmentation of Multimodal Images. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):4419–4429, 2021.

[59] Fangchang Mal and Sertac Karaman. Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, page 1–8. IEEE, 2018.

[60] Cees G. M. Snoek, Marcel Worring, Jan-Mark Geusebroek, Dennis. C. Koelma, and F. J. Seinstra. The MediaMill TRECVID 2004 Semantic Video Search Engine. In *TRECVID Workshop*, 2004.

[61] Yifei Zhang, Désiré Sidibé, Olivier Morel, and Fabrice Mériaudeau. Deep multimodal fusion for semantic image segmentation: A survey. *Image and Vision Computing*, 105:104042, 2021.

[62] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun. Indoor Semantic Segmentation using depth information. *arXiv:1301.3572*, 2013.

[63] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture. In Shang-Hong Lai, Vincent Lepetit, Ko Nishino, and Yoichi Sato, editors, *Computer Vision (ACCV)*, pages 213–228. Springer, 2017.

[64] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning Rich Features from RGB-D Images for Object Detection and Segmentation. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision (ECCV)*, pages 345–360. Springer, 2014.

[65] Zhou Guo, Rui Xu, Chen-Chieh Feng, and Zhao Zeng. PIF-Net: A Deep Point-Image Fusion Network for Multimodality Semantic Segmentation of Very High-Resolution Imagery and Aerial Point Cloud. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2023.

[66] Seungyong Lee, Seong-Jin Park, and Ki-Sang Hong. RDFNet: RGB-D Multi-level Residual Feature Fusion for Indoor Semantic Segmentation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4990–4999. IEEE, 2017.

[67] Feiyi Fang, Yazhou Yao, Tao Zhou, Guosen Xie, and Jianfeng Lu. Self-Supervised Multi-Modal Hybrid Fusion Network for Brain Tumor Segmentation. *IEEE Journal of Biomedical and Health Informatics*, 26(11):5310–5320, 2022.

[68] Grzegorz Rozenberg. *Handbook of Graph Grammars and Computing by Graph Transformation*. WORLD SCIENTIFIC, 1997.

[69] Wei Dong, Charikar Moses, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th International Conference on World Wide Web (WWW)*, WWW '11, page 577–586. ACM, 2011.

[70] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 2008.

[71] Karel Breiter, Hilton Tulio Costi, Michaela Vašinová Galiová, Michaela Hložková, Jindřich Kynický, Zuzana Korbelová, and Marek Dosbaba. Trace element composition of quartz from alkaline granites – A factor supporting genetic considerations: Case study of the Pitinga Sn–Nb–Ta–Th-cryolite deposit. *Journal of South American Earth Sciences*, 119:104025, 2022.

[72] Ludwig Reimer. *Scanning Electron Microscopy: Physics of Image Formation and Microanalysis*. Springer, 1998.

[73] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS)*. ACM, April 2024.

[74] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[75] William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019.

[76] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[77] Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. TorchMetrics - Measuring Reproducibility in PyTorch, February 2022.

[78] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[79] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Allen Kelsey, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261*, 2018.