Segmentation of magnetic data using inverse Convolution Neural Networks*

Xiaojin Tan [†a] and Eldad Haber [‡ a]

[a]Department of Earth and Ocean Science, The University of British Columbia, Vancouver, BC, Canada [§]

July 9, 2020

Abstract

Segmentation of magnetic data is the process of dividing a magnetic image data into multiple geological units. This process is typically done manually by experts, which is time consuming and inefficient. In recent years, machine learning techniques such as Convolutional Neural Networks (CNNs) have been used for semantic segmentation. Semantic segmentation is the process that associates each pixel in a natural image with a labeled class. When attempting to use the similar technology for magnetic segmentation there are a number of challenges to consider in particular, data uncertainty resulting from inconsistency, scarcity and complexity of magnetic measurements. To overcome these challenges, we develop a new process that we call magnetic semantic segmentation (MSS). This process addresses (i) the pre-processing of magnetic data in order to enable learning; (ii) the enrichment of the data set (data augmentation) by using a geostatistical technique, referred to as Multiple-Point Simulations (MPS); and (iii) the training of such a data set based on a new neural network architecture called inverse Convolution Neural Network (iCNN). As demonstrated by the results on a field magnetic data set, this approach shows its competitiveness with human segmentation and indicates promising results.

Keywords— geological mapping, convolutional neural networks, data augmentation

1 Introduction

Airborne magnetic data is frequently recorded in order to learn about different geological units and structural geology [46]. Since different rocks have different magnetic signatures, changes in the

*The code is freely available online at `https://github.com/tanxiaojin/magSemanticSegAlgorithm.git`

†Corresponding Author: xtan@eoas.ubc.ca

‡eldadhaber@gmail.com

§X.T. carried out the experiment, analyzed the data, and wrote the manuscript; E.H. conceived of the presented idea, wrote the manuscript and supervised the project; Both authors read and approved the final manuscript.

magnetic fields typically represents changes in geological units. Nonetheless, magnetic signals can be complex and therefore, mapping rock units from a magnetic map is typically done manually. Segmentation of magnetic data is the process of dividing a magnetic image data into multiple geological units. The goal of magnetic segmentation is to extract relevant geological information from magnetic data, leading to a solid-geology map, which is typically used in regional geological studies and further mineral exploration [31].

Segmentation of magnetic data can be viewed as an essential part of a geological interpretation. Experienced geologists may spend a considerable amount of time and effort to determine the distribution and geometry of magnetic sources and then identify lithostructural units and structural features from magnetic data. This manual process of interpretation is often subjective and time-consuming.

In recent years, similar tasks have been undertaken by Artificial Intelligence (AI) and, in particular, by Convolution Neural Networks (CNNs). For segmentation of natural images, this process has been named semantic segmentation and has been studied in the last few years [32, 4, 55, 9, 30, 57, 37, 10] and similar work has been done for medical images [25, 44, 35]. However, while such technology is common in other fields, it is still lacking for geological segmentation of magnetic data. It is worth noting that there are a couple of studies on applying AI to generating geological maps from geophysical datasets. [8, 18, 38, 13] attain the goal by using unsupervised learning, which do not consider any human-labeled geophysical data. [29, 19] achieve this by using supervised classification methods, which treat each pixel of the geophysical image data as a sample. Unlike those methods above, we use supervised image segmentation methods and treat patches rather than pixels as our samples, which takes the spatial relationship between neighboring pixels into account.

There are a number of challenges to consider when attempting to use the similar technology for magnetic segmentation. First, magnetic data is recorded in different ways and processed differently. Since AI systems learn by example, one has to unify the magnetic data in a consistent way. Second, while there are plenty of natural images, there are far less interpreted magnetic images, which makes data scarce. Thirdly, the relationship between magnetic data and their corresponding lithostructural units is far more complex than that between natural images and their pixel-wise labels. Any 4-year-old child, for example, is quite likely to assign a label to an image of a dog, however, different geologists often interpret magnetic maps differently.

In this paper we develop a new process that we call magnetic semantic segmentation (MSS). There are two possible routs one can take for MSS. The first is to use a massive amount of data that is collected and annotated under many geological conditions, with the aim to have an AI system that can work for most magnetic data. While this process is desired in principle it is difficult to apply in practice. This is because annotated magnetic data is scarce and furthermore, geological environments tend to be very heterogeneous which makes the development of such a system difficult. A second approach that we take here is adapting the AI system to the geological environment at hand. We envision a geologist segmenting a small piece of the data and the AI system learned

from the geologist and segmenting the rest of the region. In this way the geologist has more control over the system and its biases. This process addresses the pre-processing of magnetic data in order to enable learning, and, in particular, the enrichment of the data set (data augmentation) in a consistent manner and finally, the training of such a data set in order to imitate experts in their geological segmentation process.

The paper is structured as follows. In Section 2 we describe the magnetic data and its pre-processing. In Section 3 we discuss data augmentation by using a geo-statistical technique, referred to as Multiple-Point Simulations (MPS). In Section 4 we formulate the magnetic segmentation problem and present the learning processing as an optimization problem. In Section 5 we propose a new neural network architecture we call inverse Convolution Neural Networks (iCNN) that is specifically developed to identify patterns. Experimental results are demonstrated in section 6. Finally, in section 7, we summarize the paper.

2 Magnetic data and its pre-processing

Airborne magnetic data is one of the most common geophysical data that is collected [46]. In most systems, a magnetometer is being carried in some sort of an airborne system, measuring the total magnetic field above the earth. The data that is being recorded can be then processed in various forms before interpretation. Although magnetic data interpretation is very common [**?**, 40] We have described here the important points to be addressed for the use AI based segmentation. We do not discuss common processing steps such as dealing with cultural noise or diurnal correction but rather discuss the points that have direct impact on the training of an AI system.

While AI systems can learn from various types of data, they can be confused by inconsistent data that leads to erroneous results. Therefore, one has to be consistent when considering the input for such an AI system. Magnetic data is being flown at different heights, in different latitudes and at different resolution. One has to treat the data carefully if it is to be used in an AI system. To demonstrate the difficulties, consider the response of a single magnetic cylinder, stationed vertically, buried in a non-magnetic background at two latitudes, of 60^o and 25^o, plotted in Figure 1[1].

While both correspond to the same object, the resulting magnetic signature is very different. In order to have a consistent image, that is not influenced by the measurement location we first reduce the magnetic data to pole [46]. Reduction to pole projects removes the dependence of magnetic data on the magnetic inclination, converting data that have been recorded in the inclined Earth's magnetic field to what they would have looked like if the magnetic field had been vertical.

A second issue to consider when using magnetic data is its height above the surface and resolution. These two items are intrinsically linked. For data that are recorded closer to the surface with platforms that move slowly, more details are revealed however, as the flight height increases, and the speed of flight is increased, less details are visible, sharp features become broader and the

[1]Figure is taken from `http://hs.umt.edu/geosciences/faculty/sheriff/courses/439-applied-magnetics/default.php`

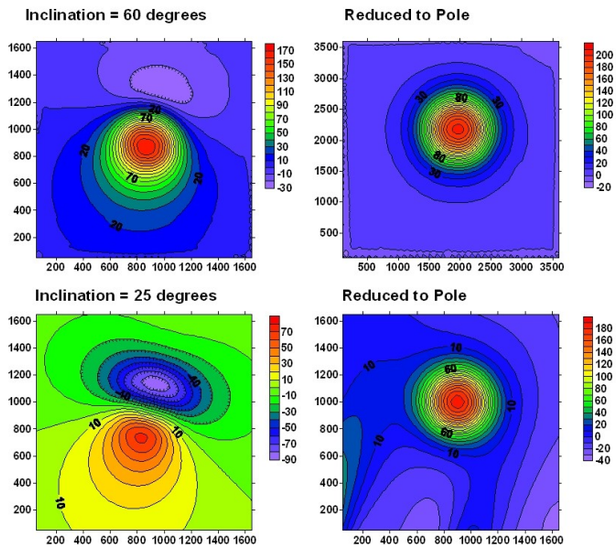


Figure 1: The magnetic data that results from a cylinder at two different latitudes and its reduction to pole.

magnetic data may look rather different.

If we are to learn from the data, we need it to be with a similar statistics and scale. Learning from data at one scale and inferring on a different scale may lead to gross inaccuracies. In order to obtain similar scales we use the process of analytical continuation [5] that uses the magnetic potential in order to transfer the data recorded at one height to a different one. This enables us to obtain consistent magnetic data with respect to both latitude and flight of height.

A third processing step that is commonly done is the computation of vertical derivatives [5]. Vertical derivatives are commonly output from different geophysical software. They tend to cancel some of the regional bias (remove the DC shift) as well as shift the spectrum of the data and are used in manual segmentation. Here we use these derivatives and output an additional map of derivatives to be used by the AI system. It is important to note that other filters can be applied to the data and be used for the training. In our experiments, we did not find that such filters improve the results. Choosing filters is often refers to as feature engineering and it is an important research topic in many different fields.

3 Data Augmentation

As fluxgate and cesium vapor magnetometers and GPS navigation techniques become popular, rapid and inexpensive acquisition of large scale magnetic datasets becomes practical. Geological interpretation of such datasets, however, requires continuous and tedious efforts by experienced geologists. Since geological environments tend to be highly heterogeneous, we need many annotated examples. This makes the amount of available geological maps (pixel-wise labels) inadequate and insufficient for the purpose of machine learning.

For machine learning problems such as automatic segmentation of magnetic data, the sufficient number of annotated magnetic images available for training is a central factor in improving perfor-

mance and reducing overfitting. Since large public data sets of segmented data are not generally available, data augmentation techniques, that is, synthetically creating new samples from original training data, are commonly employed to enlarge the size of the training dataset and avoid poor generalization. Traditional data augmentation methods [54] for machine learning tasks generate additional samples from the original training dataset by simple techniques such as mirroring, random cropping, rotation, shearing and color shifting [28]. While achieving some success [28] in many computer vision tasks, these simple techniques have some limitation in augmentation of practical magnetic data.

First, some of these simple techniques such as shearing and color shifting may fail to take the spatial characteristics of magnetic data into account. In contrast to benchmark datasets such as CIFAR-10 [27] and ImageNet [11], practical magnetic data reveal the inhomogeneous distribution of magnetic minerals and the complex geometry of magnetic sources, which can not be preserved and reproduced by some of these traditional data augmentation techniques; second, even if some of these image transform operations (e.g. mirroring, cropping and rotation) could preserve the shape of objects [7] [24] [42], they were unfit for use in our practical magnetic data. The cause lies in the fact that, in most cases, only a few or perhaps even one annotated magnetic image is available. These geometry-preserving transforms and their various combinations can not generate sufficient number of labeled datasets to prevent overfitting in convolutional neural networks.

These observations lead us to a key question concerning augmenting magnetic data: How to generate adequate number of geologically feasible datasets if only a few or perhaps even one annotated magnetic image is available? To address this question, we propose a solution to magnetic data augmentation tasks based on Multiple-Point Simulations (MPS) [14] [1] [56] [21]. Using this method, we are able to generate sufficient (in theory an infinite) number of new samples based on the patterns and distributions of heterogeneities (multiple-point statistics) inferred from the original data.

3.1 Data Augmentation based on MPS

MPS is a geostatistical simulation technique widely used to model complex geological patterns and subsurface heterogeneity (for a detailed review of MPS see, e.g., [23] [34] and references therein). There have been a great variety of implementation methods developed within MPS. Each of them has certain limitations and their applicable domains largely depend on the characteristics of the training image, the nature of conditioning data, the computational time desired, and other practical considerations [33]. MPS can be used with a single image, yielding many images with "similar statistics". It is important to recall that if the geological conditions are dramatically different than one needs to obtain more than a single image in order to generate augmented images that result from different statistics.

In this section, we will review MPS [14] [1] [56] [21] [22] by briefly describing our magnetic data as training images and the characteristics of data augmentation tasks relevant to MPS, aiming at

selecting a suitable implementation algorithm for our magnetic data augmentation problem. To avoid confusion with the concept "training data" defined in the context of supervised learning, "exemplars" [39] that is the computer graphics word for training images will be used below to replace "training images"

- We denote by $\mathbf{ex}$ an exemplar consisting of b pixels which are β-dimensional feature vectors,

$$\mathbf{ex} \in \mathbb{R}^V \qquad V = b\beta \tag{3.1}$$

in which $\beta = 1$ for a grayscale exemplar or $\beta = 3$ for an RGB exemplar. Note that $\mathbf{ex}$ can also be multivariate. Such multivariate $\mathbf{ex}$ consists of more than one type of images which will be discussed and dealt with in detail later.

For most applications of MPS, the generated realizations need to be conditioned to various sources of measured data (e.g. well-log data) to better capture the subsurface structures of a field. This is called conditional MPS, which can be formulated as,

$$\mathbf{y} = S_c(\mathbf{ex}, \mathbf{d}) \tag{3.2}$$

where we assume the realization, denoted by $\mathbf{y} \in \mathbb{R}^V$, and the conditioning data, denoted by $\mathbf{d} \in \mathbb{R}^V$, have the same size as the original exemplar; a conditional single-grid simulation, denoted by S_c, represents a function generating the realization $\mathbf{y}$ conditioned to the data $\mathbf{d}$ using the exemplar $\mathbf{ex}$. Contrarily, our task of magnetic data augmentation does not need to integrate any measured data. As a result, our task can be viewed as an application of unconditional MPS simulation, which can be formulated as,

$$\mathbf{y} = S_u(\mathbf{ex}) \tag{3.3}$$

where S_u, called an unconditional single-grid simulation, represents a function generating the realization $\mathbf{y}$ using the exemplar $\mathbf{ex}$, without needing data conditioning.

- Typically, large-scale structures (e.g. long range continuous channels) are presented in our exemplars. To capture and reproduce the long range continuity of the spatial structures, a **multi-scale technique**, called multi-resolution simulation [6] [52] [21] [47] will be incorporated into our data augmentation task. This technique first creates a pyramid of multiple resolution views of the exemplar and then construct realizations in a coarse-to-fine fashion at each resolution level such that each finer resolution level is constructed from the already simulated coarser levels. It involves downsampling of the exemplar as well as upsampling of the realizations. In terms of mathematical notation, we denote the exemplar at multi-resolution level of r by $\mathbf{ex}^{(r)}$ where $r = 1, \ldots, n$ and let the original exemplar $\mathbf{ex} = \mathbf{ex}^{(1)}$. According to the multigrid literature [47], downsampling to a smaller image $\mathbf{ex}^{(r+1)}$ from a

higher-resolution image $\mathbf{ex}^{(r)}$ can be represented as,

$$\mathbf{ex}^{(r+1)} = \mathbf{R}\mathbf{ex}^{(r)} \quad \text{for} \quad r = 1, \ldots, n-1. \tag{3.4}$$

where $\mathbf{R}$ is a restriction matrix that linearly transforms (e.g. averages) the higher-resolution image $\mathbf{ex}^{(r)}$ into the lower-resolution image $\mathbf{ex}^{(r+1)}$. Figure (2) illustrates downsampling of the continuous channelized images with three multi-resolutions. Similarly, upsampling from the realization at multi-resolution level of r, denoted as $\mathbf{y}^{(r)}$, to a higher resolution image, denoted as $\mathbf{y}^{(r-\frac{1}{2})}$, can be represented as,

$$\mathbf{y}^{(r-\frac{1}{2})} = \mathbf{P}\mathbf{y}^{(r)} \quad \text{for} \quad r = n, \ldots, 2. \tag{3.5}$$

where the coarsest resolution of the realization $\mathbf{y}^{(n)}$ is generated using single-grid approach with the coarsest resolution of the exemplar $\mathbf{ex}^{(n)}$,

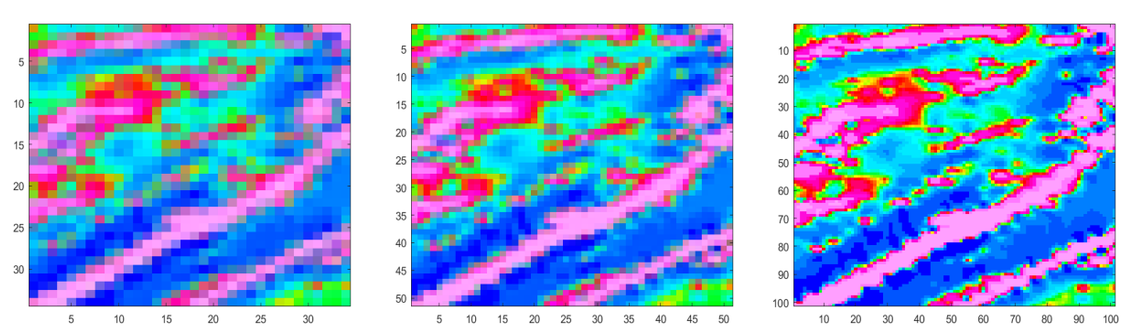
$$\mathbf{y}^{(n)} = \mathbf{S}_u(\mathbf{ex}^{(n)}) \tag{3.6}$$

and $\mathbf{y}^{(r-\frac{1}{2})}$ can be viewed as conditioning data to construct the realization $\mathbf{y}^{(r-1)}$ at resolution level of $r-1$,

$$\mathbf{y}^{(r-1)} = \mathbf{S}_c(\mathbf{ex}^{(r-1)}, \mathbf{y}^{(r-\frac{1}{2})}) \quad \text{for} \quad r = n, \ldots, 2. \tag{3.7}$$

Here, $\mathbf{P}$ is a prolongation matrix that uses interpolation (e.g. bicubic) to transform images from lower to higher resolution; The single-grid simulations $\mathbf{S}_u$ and $\mathbf{S}_c$ will be described in detail in the next paragraph.

It is noteworthy that the creation of multiple resolution views relies on the interpolation method (e.g. bicubic) by which the output pixel value is a weighted average of pixels in the nearest neighborhood. For a categorical variable, the weighted average of neighboring pixels becomes no longer categorical. In such cases, Otsu's method [36] is employed to change the weighted average of values back to be categorical.



(a) coarsest image size : 34×34 (b) coarser image size : 51×51 (c) original image size : 101×101

Figure 2: a channelized image and its multiple resolution views

- Depending on how nodal values in realizations are simulated, MPS algorithms can be classified

roughly into two categories: pixel-based and patch-based [56] [1] [12] [45] methods.

- pixel-based methods typically focus on estimating the node-specific conditional probability distribution derived from the exemplars in a sequential mode. The conditional probability for each node $\mathbf{u}$ can be formulated as follows,

$$F(\mathbf{z}) = P(Z(\mathbf{u}) \leq \mathbf{z} | B(\mathbf{u})) \tag{3.8}$$

  where $Z(\mathbf{u})$ is the random variable to be simulated; $B(\mathbf{u})$, termed data event, consists of previously simulated values in the neighborhood of the node $\mathbf{u}$. To evaluate the equation (3.8), the exemplars are scanned to extract all the replicates of the data event, whose central values are recorded to build an empirical conditional probability distribution of $Z(\mathbf{u})$ from which a sample is drawn to simulate the nodal value.

- Instead of sampling a single nodal value from the conditional probability, patch-based methods paste an entire patch directly onto the simulation grid, using the concept of distance (or similarity) between the data event of the simulation grid and the data events extracted from the exemplar, which can be denoted by $d(B(\mathbf{u}_1), B(\mathbf{u}_2))$. A simple example of a distance function that often works well for continuous variables is the Euclidean distance, expressed as,

$$d(B(\mathbf{u}_1), B(\mathbf{u}_2)) = \frac{1}{\rho} \sum_{i=1}^{\rho} \left\| \mathbf{B}_i^1 - \mathbf{B}_i^2 \right\|^2 \tag{3.9}$$

  where $\mathbf{B}^1$ and $\mathbf{B}^2$ are abbreviated from $B(\mathbf{u}_1)$ and $B(\mathbf{u}_2)$ respectively; ρ is the total number of nodes in these data events. For categorical variables, a simple measure of similarity can be represented as,

$$d(\mathbf{B}^1, \mathbf{B}^2) = \frac{1}{\rho} \sum_{i=1}^{\rho} \left\| \mathbf{B}_i^1 - \mathbf{B}_i^2 \right\|^0 \tag{3.10}$$

  which indicates the percentage of ill-matched nodes. It is noteworthy that the pattern recognition literature [3] provides many other different distance functions, which should be selected properly according to the property of our variables.

Pixel-based methods generate realizations pixel by pixel, which makes the conditioning to various sources of measured data much easier than in patch-based methods; patch-based methods, however, generally have better pattern reproduction and faster simulation speed since they synthesize patches rather than pixels [51]. As discussed before, our task of magnetic data augmentation does not have any measured data to be conditioned. As a result, patch-based methods will be integrated into our task and the single-grid unconditional simulation S_u using patch-based methods can be summarized as follows,

1. scanning the exemplar **ex** with a fixed template to extract a list of patches (multi-point

vectors), denoted as data events $\mathbb{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_t\}$, where t is the number of patches extracted from **ex**.

2. define a random path on the simulation grid
3. for each node $\mathbf{u}$ of the simulation grid along the random path, we calculate a distance (mismatch) between the data event $B(\mathbf{u})$, formed by the previously simulated nodes in the patch centered on $\mathbf{u}$, and the data event $\mathbf{p}_i$ in $\mathbb{P}$,

$$\min_i \quad d(B(\mathbf{u}), \mathbf{p}_i). \tag{3.11}$$

Once the most similar patch in $\mathbb{P}$ is found, we paste the patch into the simulation grid to replace the current data event. Such distance in general is (3.9) for continuous variables and (3.10) for categorical cases.

4. construct a realization $\mathbf{y}$ by proceeding to next node until all nodes of the grid have been simulated
5. note that a number of different realizations can be generated by using different paths on the simulation grid.

In the next two paragraphs, new types of distance functions will be considered to deal with multivariates and nonstationarity, which are commonly found in annotated magnetic data.

- Unlike the univariate case, annotated magnetic data treated as exemplars, consist of a combination of several continuous variables and categorical variables, which are collocated and spatially interdependent. One of the major aims of magnetic data augmentation is to generate realizations of these variables such that (1) each variable in realizations shares the same spatial structure with that in exemplars; (2) these variables in realizations are physically consistent with each other. For this purpose, the concept of multivariate data event, consisting a combination of continuous and categorical variables, needs to be developed and distance functions specially designed for these data events are needed. Generally, a distance between multivariate data events, denoted by $D(\mathbf{B}^1, \mathbf{B}^2)$, can be defined as a weighted average of the distances between each variable of each data event,

$$\begin{aligned} D(\mathbf{B}^1, \mathbf{B}^2) &= \sum_{k=1}^{M} w_k d_k(\mathbf{B}^{1_k}, \mathbf{B}^{2_k}) \\ \text{s.t.} \qquad & \sum_{k=1}^{M} w_k = 1 \end{aligned} \tag{3.12}$$

where M is the total number of variables contained in the data event; w_k indicates the relative importance of the k_{th} variable; $\mathbf{B}^{1_k}$ represents the k_{th} variable of the first data event; The distance d_k, defined by the equation (3.9) and (3.10), is the distance for the k_{th} variable of the data events. Clearly, the distance defined in the equation (3.12) can be easily integrated

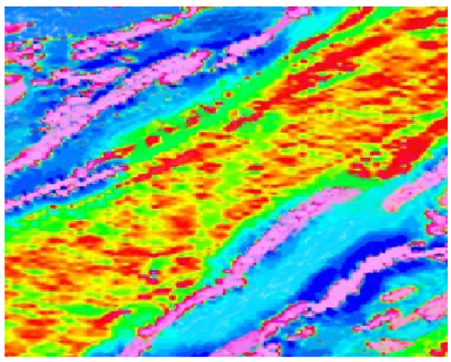


Figure 3: Example of nonstationarity based on our magnetic image

into our task, in which annotated magnetic data consist of both continuous and categorical variables.

- Magnetic patterns, defined by the intensity and geometry of magnetic sources from our magnetic data, are often complex and tend to vary in space, meaning that they are statistically different over the entire spatial domain. Figure (3) shows that the types of geological patterns, consisting of channelized and non-channelized structures, vary over the spatial domain.

During the simulation, data events extracted from the exemplars are compared with the data event of the simulation grid based on the equation (3.12), such that the data event from the exemplars, most similar to the data event of the simulation grid, will be selected and pasted onto the simulation grid. Such a distance is used under the assumption of stationarity and fails to consider the location of these data events. To counteract this problem, the spatial grid location of each data event from the exemplars should be used. Specifically, a Euclidean distance between spatial locations of data events can be expressed as follows,

$$d(\mathbf{u}_1, \mathbf{u}_2) = \|\mathbf{u}_1 - \mathbf{u}_2\|^2 \tag{3.13}$$

By combining (3.12) with (3.13), a new distance [22] for nonstationary modeling, denoted by $D_n(B(\mathbf{u}_1), B(\mathbf{u}_2))$, is proposed to account for not only the similarity between the data events, but also the spatial distance between them,

$$D_n(B(\mathbf{u}_1), B(\mathbf{u}_2)) = (1-\delta)D(\mathbf{B}^1, \mathbf{B}^2) + \delta d(\mathbf{u}_1, \mathbf{u}_2) \tag{3.14}$$

where the weights δ can be modulated to determine the non-stationarity level of the simulation.

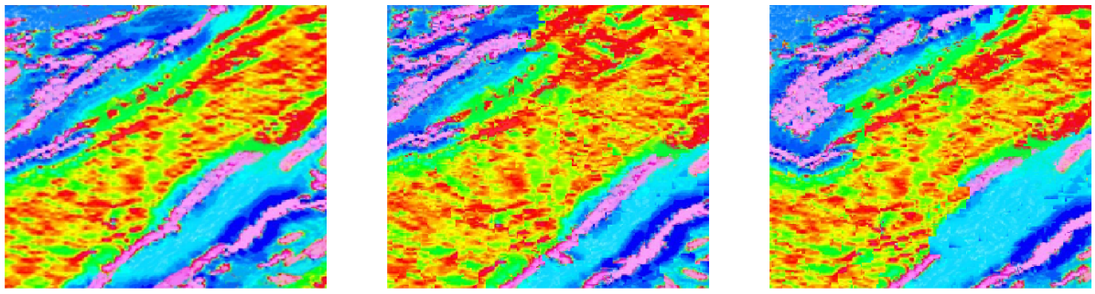
In summary, our task of magnetic data augmentation can be described as follows,

Figure (4) shows a simplified example of generating two different realizations with our method summarized above, using one type of magnetic images as the exemplar. These generated realizations

Algorithm 1: Magnetic Data Augmentation

[1] create annotated magnetic data $\mathbf{ex}$ with n multi-resolutions based on the equation (3.4), where we denote the magnetic data at multi-resolution level of r by $\mathbf{ex}^{(r)}$
[2] set the random seed to a constant value
[3] generate the coarsest resolution of the realization $\mathbf{y}^{(n)}$ using unconditional single-grid approach with the coarsest resolution of the magnetic data $\mathbf{ex}^{(n)}$ (see the equation (3.6))
[4] set $r = n - 1$
[5] generate $\mathbf{y}^{(r)}$ using conditional single-grid simulation, with $\mathbf{ex}^{(r)}$ and the realization $\mathbf{y}^{(r+\frac{1}{2})}$ upsampled from the lower-resolution realization $\mathbf{y}^{(r+1)}$ (see the equation (3.7))
[6] set $r = r - 1$ and repeat step 5 until $r = 1$ to generate the desired output $\mathbf{y}^{(1)}$
[7] change the random seed and repeat steps $3 - 6$ until the desired number of realizations is reached.

will be used as input in our magnetic data segmentation task, which will be discussed in detail in the next sections.



(a) an exemplar (b) realization 1 (c) realization 2

Figure 4: a simplified example of generating two different realizations with our method summarized above, using one type of magnetic images as the exemplar

4 Magnetic Segmentation Problem setup

The image segmentation problem, also called the pixel-wise classification problem, can be stated as follows: We denote by $\mathbf{\Theta}$ an image consisting of b pixels which are β-dimensional feature vectors,

$$\mathbf{\Theta} = \{\mathbf{x_1}, \ldots, \mathbf{x_b}\} \in \mathbb{R}^{\beta \times \mathbf{b}} \tag{4.15}$$

where the rows of the image $\mathbf{\Theta}$ are stacked into a long vector $\mathbf{y} \in \mathbb{R}^V$, as defined in Section 3.

Similarly, we denote its corresponding pixel-level labels by $\mathbf{c}$,

$$\mathbf{c} = \{c_1, \ldots, c_b\} \in \mathbf{L}^b \tag{4.16}$$

where $\mathbf{L} = \{l_1, \ldots, l_\kappa\}$ is the label space and κ is the total number of classes. In practice, the pixel-level labels $\mathbf{c}$ are often represented using probability maps $\mathbf{\Phi} = \{\mathbf{q}_1, \ldots, \mathbf{q}_b\}$ in which $\mathbf{q}_i$ is a κ-dimentional vector whose element q_{ij} represents the likelihood of a pixel j belonging to class l_i and must meet the following criteria,

$$q_{ij} \in [0, 1], \quad \sum_{i=1}^{\kappa} q_{ij} = 1 \tag{4.17}$$

Specifically, the probability q_{ij} is defined as follows,

$$q_{ij} = \begin{cases} 1 & \textbf{if} \quad c_j = l_i \\ 0 & \textbf{otherwise} \end{cases}$$

We consider the image segmentation problem in which a training set $\mathbf{D}$ consisting of s training examples will be used for training,

$$\mathbf{D} = \{\mathbf{Y}_0, \mathbf{Q}_0\} = \{(\mathbf{y}^{(i)}, \mathbf{Q}^{(i)}); \quad i = 1, \ldots, s\} \tag{4.18}$$

where $\mathbf{Y}_0 \in \mathbb{R}^{V \times s}$ and $\mathbf{Q}_0 \in \mathbb{R}^{\kappa \times b \times s}$

In the context of magnetic segmentation, the magnetic data can produce many types of images, each of which enhances different geological features. Based on this observation, β depends on the number of types of images generated from magnetic data as well as the number of color channels of each image, and κ is the number of geological units. The goal is to learn some mapping function $g(\mathbf{Y}_0)$ that approximates the data-label relation such that it generalizes well and can be used to specifies a categorical label for each pixel in new unlabeled images. In general the mapping function $g(\cdot) : \mathbb{R}^{V \times s} \to \mathbb{R}^{\kappa \times b \times n}$ can be anything. In this paper, the mapping function $g(\cdot)$ will be modeled and parametrized by a new deep network architecture called iCNN which will be discussed in detail in section 5.

Having introduced the form of the mapping function that approximates the highly nonlinear relationship between the data and the labels, the problem of magnetic segmentation can boil down to an optimization problem, aiming at estimating the parameters of the mapping function that explains the data-label relation and generalizes well to similar unlabeled magnetic data. It can be formulated as follows,

$$\underset{\boldsymbol{\theta}, \mathbf{W}}{\text{minimize}} \quad \frac{1}{s} \sum_{i=1}^{s} J(g(\mathbf{y}^{(i)}; \boldsymbol{\theta}), \mathbf{Q}^{(i)}; \mathbf{W}) + \alpha R(\boldsymbol{\theta}, \mathbf{W}) \tag{4.19}$$

where the loss function J evaluates the cost of predicting $g(\mathbf{y}^{(i)}; \boldsymbol{\theta})$ when the ground truth labels are $\mathbf{Q}^{(i)}$; the regularizer $R(\boldsymbol{\theta}, \mathbf{W})$ is typically chosen to penalize undesirable parameters; the parameter $\alpha > 0$ controls the relative weighting between the twin goals of minimizing the data fit and preventing overfitting. There are several ways to define the details of the loss function. Since we are working with image segmentation problems, the cross-entropy loss function, defined for each pixel, will be used,

$$J(g(\mathbf{y}; \boldsymbol{\theta}), \mathbf{Q}; \mathbf{W}) = -\mathsf{trace}(\mathbf{Q}^T \log \mathbf{Q}_p(\boldsymbol{\theta}, \mathbf{W})) \tag{4.20}$$

where

$$\mathbf{Q}_p(\boldsymbol{\theta}, \mathbf{W}) = \exp(\mathbf{W} g(\mathbf{y}; \boldsymbol{\theta}))\mathsf{diag}\left(\frac{1}{\mathbf{e}^T \exp(\mathbf{W} g(\mathbf{y}; \boldsymbol{\theta}))}\right) \tag{4.21}$$

here, the parameters $\boldsymbol{\theta}$ will be defined in the next section, the classifier $\mathbf{W} \in \mathbb{R}^{\kappa \times \kappa}$ and $\mathbf{e} =$

$(1, 1, \ldots, 1)^T \in \mathbb{R}^\kappa$; the exponential and the division operations are applied element-wise.

5 Neural network architectures

5.1 The diffusion reaction network

Image segmentation using neural networks has been successful for many types of natural images [32, 4, 55, 9, 30, 57, 37, 10] and medical images [25, 44, 35]. Here we focus our attention on a residual network (Resnet) [20] that have favorite stability properties [16]. Residual networks have been very successful for problems of image classification and recently have been used for image segmentation [9, 10].

A classical Resnet can be written as

$$\mathbf{Y}_{j+1} = \mathbf{Y}_j + hf(\mathbf{Y}_j, \boldsymbol{\theta}_j) \tag{5.22}$$

Here, the parameters $\boldsymbol{\theta} = \{\mathbf{K}, \mathbf{b}\}$ and f is a double layer of the form

$$f(\mathbf{Y}, \boldsymbol{\theta}) = -\mathbf{K}^\top \sigma(N(\mathbf{KY}) + \mathbf{b}) \tag{5.23}$$

Here, $\mathbf{K}$ is a convolution matrix, $\mathbf{b}$ is a bias vector and σ is an activiation function. The function N is a normalization layer

$$N(\mathbf{Z}) = \frac{\mathbf{Z}}{\sqrt{\sum_j \mathbf{Z}_j^2 + \epsilon}} \tag{5.24}$$

The sum in (5.24) can be taken over the training data which leads to a batch norm or over the channels of each data point which leads to instance norm (see [49] for details).

While the Resnet architecture works well it has one main disadvantage. At each step of the Resnet each pixel talks only to its neighbors in each channel of the image [15]. This has lead researchers to change the resolution of the network and combine the different resolutions together through a so called U-net [41]. The U-net, as discussed in [15], has more complex architecture than simple Resnets, is less well-understood theoretically, and usually requires many more parameters. Since geological data typically contains many scales, we propose a different type of network motivated from pattern formation [48] [53] that can reproduce natural occurring patterns.

It has been discussed in [16] that the classical Resnet can be viewed as a forward Euler discretization of the ODE

$$\mathbf{Y}_t = -\mathbf{K}(t)^\top \sigma(N(\mathbf{K}(t)\mathbf{Y}) + \mathbf{b}(t)).$$

This system can be viewed as a diffusion system that has favorite properties for filtering. This type of network can enhance local features and help in the classification of an image, however, this type of equation tends to lose local information (due to diffusion) which may be important for pixel-wise

classification. In [15], a different network can be obtained by considering a different continuous model of the form

$$\mathbf{Y}_t = -\mathbf{K}(t)^\top \mathbf{K}(t)\mathbf{Y} + \sigma(N(\mathbf{S}(t)\mathbf{Y}) + \mathbf{b}(t)) \tag{5.25}$$

Here, $\mathbf{K}$ is a diagonal convolution operator that is

$$\mathbf{K}(t) = \begin{pmatrix} \mathbf{K}_1(t) & & \\ & \ddots & \\ & & \mathbf{K}_m(t) \end{pmatrix}. \tag{5.26}$$

and $\mathbf{S}$ is a 1D convolution, that is, it couples the different channels but does not couple any pixels in space. Note that the diagonal convolution appears linearly in the equations and couples only pixels in space while the 1D convolution does not couple any spatial features but does couple the different channels.

This model is referred to as Diffusion-Reaction equation and is commonly used for problems of pattern formation [48] [53]. The first term $-\mathbf{K}^\top\mathbf{K}$ is the diffusion term while the second term is the reaction one. Similar models have been used for the prediction of patters such as zebra stripes, butterfly wings [43], crystal formation [50] and electrical waves in the heart [26]. Since geological structures can be thought of special patterns we are motivated to use this network for our problem. A more detailed discussion about the method and its advantages comparing it to classical ResNets and U-nets can be found in [15].

Clearly, the continuous model (5.25) needs to be discretized if we are to use it for our problem. A common discretization for diffusion reaction equations is the Implicit Explicit schemes (IMEX) [2] where an explicit step on the reaction term is followed by an implicit step on the diffusion term. A simple IMEX method for our problem reads

$$\mathbf{Y}_{j+\frac{1}{2}} = \mathbf{Y}_j + h\sigma(N(\mathbf{S}_j\mathbf{Y}_j) + \mathbf{b}_j) \tag{5.27a}$$

$$\mathbf{Y}_{j+1} = (\mathbf{I} + h\mathbf{K}_j^\top\mathbf{K}_j)^{-1}\mathbf{Y}_{j+\frac{1}{2}} \tag{5.27b}$$

We refer to the network (5.27) as an IMEX network.

Before we discuss the computation of the IMEX steps let us explore some of the properties of such a system. First, and most importantly, the second step of the IMEX step is non-local, that is, any pixel in $\mathbf{Y}_{j+1}$ is impacted by the values of **all** the pixels in $\mathbf{Y}_j$. This is particularly important when considering processes that occur on many scales. Second, the implicit step is **unconditionally stable** that is, it filters increasing modes in $\mathbf{Y}_j$. The explicit step is a usual Resnet step but with only 1D convolutions. This makes this step cheap compared to the usual Resnet steps that requires at least 3×3 convolutions.

5.2 Computing the diffusion reaction network

The IMEX network (5.27) involves two steps. The first step can be computed using standard convolution tools, requiring a computational cost of $\mathcal{O}(m^2 n)$ where m is the number of feature maps as well as the number of filters and n is the number of pixels in the image; the second step is slightly more complex since it involves the inversion of a matrix, however, if we use circular boundary conditions this inversion can be done in $\mathcal{O}(mn \log n)$ [17]. Since $\log n$ is typically much smaller than the number of filters m, the increased cost of the second step is trivial [15].

When periodic boundary conditions are employed, the matrix $\mathbf{K}_i$ is referred to as Block Circulant with Circulant Blocks (BCCB), which can be decomposed as

$$\mathbf{K}_i = \mathbf{F}^* \mathbf{\Lambda}_i \mathbf{F} \tag{5.28}$$

where $\mathbf{F}$ is the two-dimensional unitary discrete Fourier transform (DFT) matrix and $\mathbf{\Lambda}_i$ is a diagonal matrix containing the eigenvalues of $\mathbf{K}_i$. This implies that

$$\mathbf{K}_j^T \mathbf{K}_j = \mathbf{F}^* (\mathbf{\Lambda}_j^* \mathbf{\Lambda}_j) \mathbf{F}. \tag{5.29}$$

And the inverse convolution matrix can be written as

$$(h\mathbf{K}^T\mathbf{K} + \mathbf{I})^{-1} = \begin{pmatrix} \mathbf{F}^* \left(\frac{1}{1+h\mathbf{\Lambda}_1^* \mathbf{\Lambda}_1} \right) \mathbf{F} & & \\ & \ddots & \\ & & \mathbf{F}^* \left(\frac{1}{1+h\mathbf{\Lambda}_m^* \mathbf{\Lambda}_m} \right) \mathbf{F} \end{pmatrix} \tag{5.30}$$

Note that since $\mathbf{\Lambda}_j$ are diagonal the division is pointwise.

The product of $(h\mathbf{K}^T\mathbf{K} + \mathbf{I})^{-1}$ with a vector can therefore be done by the following algorithm

---

Algorithm 1: Computing the implicit step $(\mathbf{I} + h\mathbf{K}^\top\mathbf{K})^{-1}\mathbf{Y}$

[1] Compute the FFT of the different channels of $\mathbf{Y}$
[2] Compute the FFT of the convolution matrix $\mathbf{K}$
[3] Pointwise divide each channel by $1 + h\mathbf{\Lambda}_j^* \mathbf{\Lambda}_j$
[4] Transform the result using the inverse FFT transform

Note that step 3 can be interpreted as a 1D convolution with a diagonal matrix and can be done using standard tools. A sketch of the iCNN architecture compared with the Resnet is plotted in Figure 5.

the details of our full neural network architectures are shown in Figure 5

6 Experiments

6.1 Magnetic images

We now show the use of our approach on field data. The field data was obtained from Computational Geoscience Inc[2] and has been processed by geologists working on a field data set.

[2] www.cgiinc.com

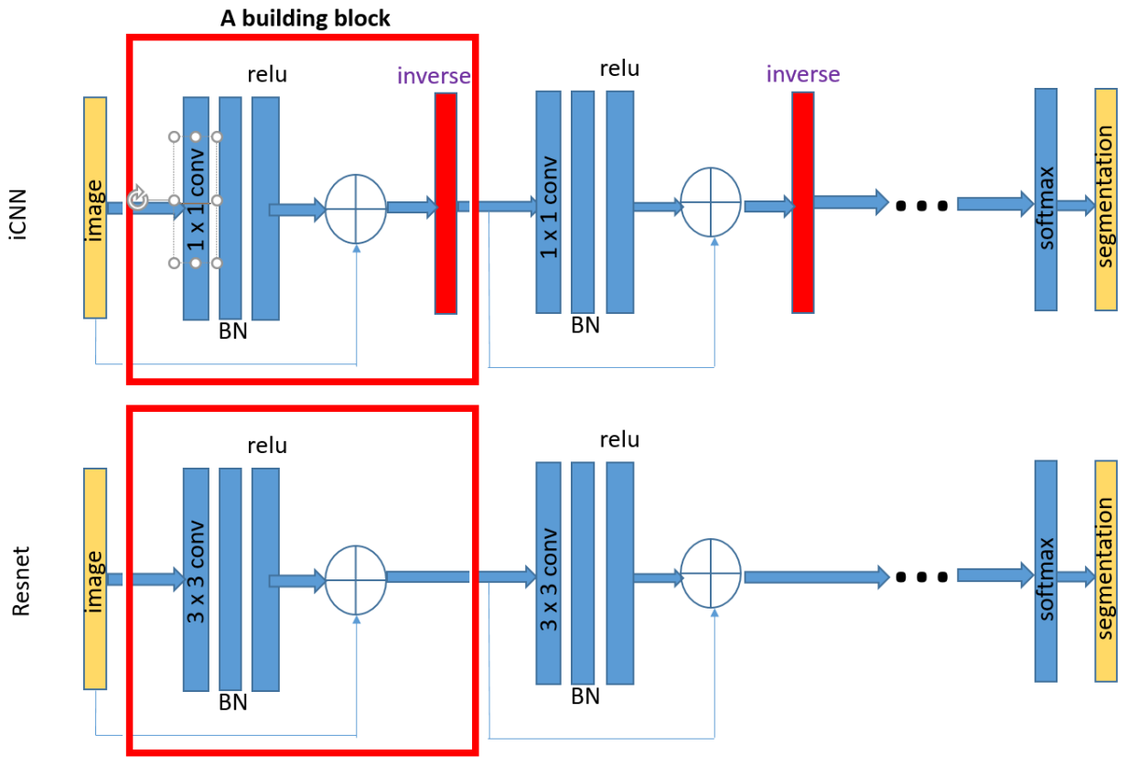
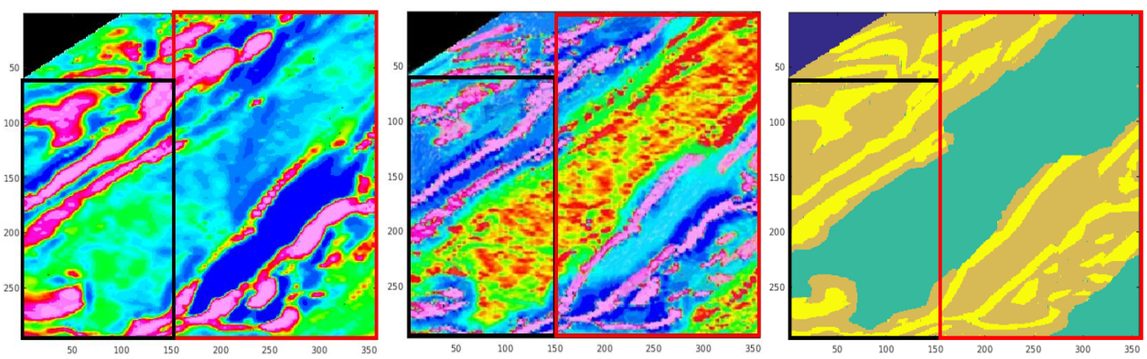


Figure 5: building blocks and architecture of Resnet and its counterpart iCNN version

The study area consists of two types of magnetic images produced from the magnetic data:

- Total magnetic intensity image (TMI) with RGB channels that shows large scale geologic features
- the First Vertical Derivative of TMI (1VD) with RGB channels that reveals detailed geologic features

Geological interpretations of the study area were provided by an expert geologist, leading to a solid geological map with 3 classes of geological units. The training area of the size 237×151 and the test area of the size 297×206 are shown in Figure 6.



(a) study area: TMI (b) study area: 1VD (c) study area: Geology

Figure 6: Black Box: Training area of the size 237×151; Red Box: Test area of the size 297×206

6.2 Magnetic Data Augmentation

In this section, we employ our MPS-based data augmentation which takes the training area as input to synthesize geologically realistic realizations. 15 realizations are generated and Figure 7 displays

two of these realizations which is geologically realistic and has the similar spatial continuity to the training area. Some parameters used for generating different realizations are listed in Table 1.

realization id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
template size	21	21	13	13	21	21	13	13	21	13	13	13	21	21	21
weight ω	0.1	0.3	0.5	0.5	0.5	0.7	0.8	0.8	0.8	0.5	0.5	0.5	0.5	0.5	0.5

Table 1: Some parameters modulated to generate different realizations

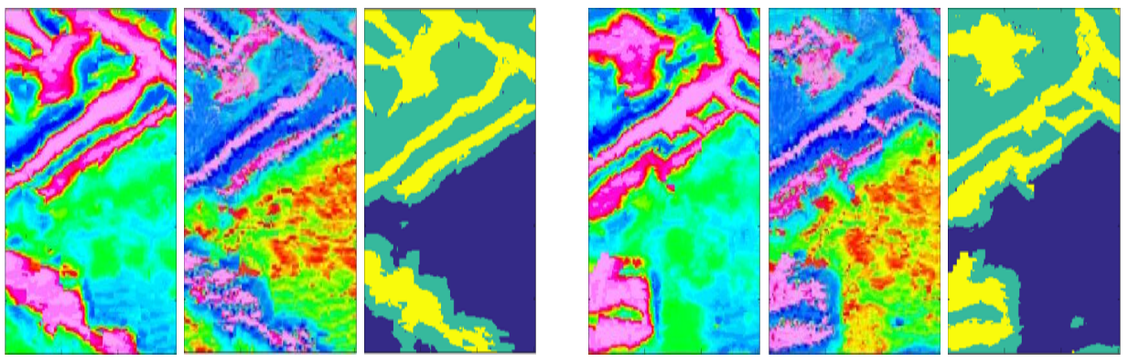
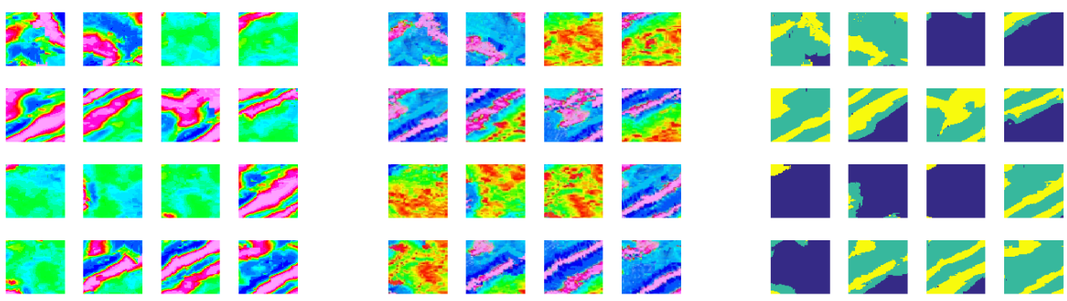


Figure 7: Left: Realization 2 with a template size of 21×21, and a non-stationary weight factor of 0.3; Right: Realization 4 with a template size of 13×13, and a non-stationary weight factor of 0.5



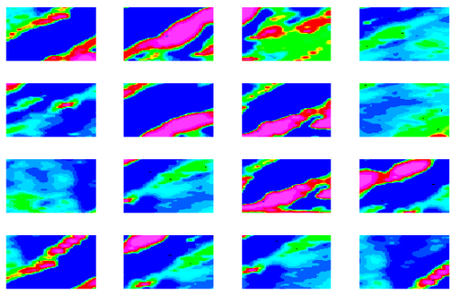
(a) training patches: TMI (b) training patches: 1VD (c) training patches: Geology

Figure 8: 16 training patches and their corresponding geological maps

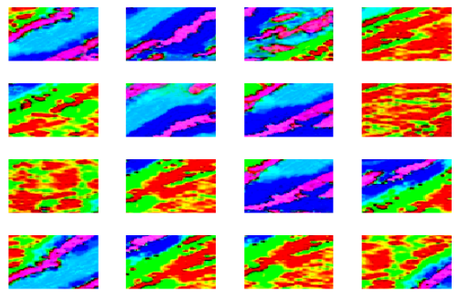
6.3 Magnetic Data Segmentation

3000 patches of the size $64 \times 64 \times 7$ are randomly sampled from the training area as well as from the 15 realizations. Each patch consists of one training sample of the size $64 \times 64 \times 6$ and its manual segmentation map of the size 64×64. Figure 8 displays 16 training patches and their corresponding geological maps. Similarly, 300 patches of the same size are randomly extracted from the test area.

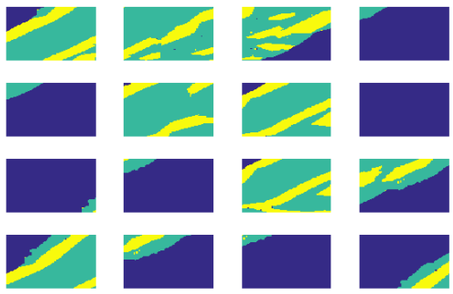
Our iCNN network consists of 3 blocks. The first block contains a convolution layer with 32 filters of the size $3 \times 3 \times 6$ with a stride of 1, which outputs 32 feature maps of the size 64×64. The second block is formed by the alternation of 2 explicit layers and 2 implicit layers: (1) The explicit layer is essentially a convolution layer which takes as input the output of the previous layer and filters it with 32 kernels of the size $1 \times 1 \times 32$; (2) the implicit layer involves an inverse of a sparse block diagonal matrix of the size 131072×131072, which will not be formed explicitly. Note that



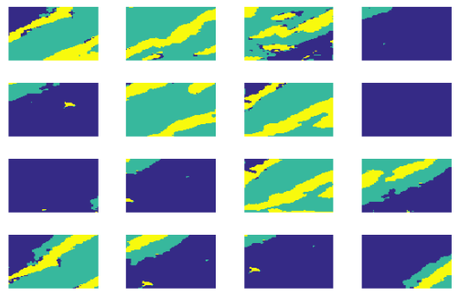
(a) test patches: TMI



(b) test patches: 1VD



(c) Manual Segmentation



(d) Automatic Segmentation

Figure 9: 16 out of 200 test patches and their manual and automatic segmentations

the size of feature maps remains the same through a stack of explicit and implicit layers in block 2. The last block contains a pixel-wise soft-max layer, which outputs the normalized probabilities of each pixel for each class.

We also train its ResNet counterpart. Both of these two networks are the same except that (i) the ResNet does not have implicit layers; (ii) the explicit step of ResNet utilizes 3×3 convolutions.

We demonstrate the application of iCNN and its ResNet counterpart to our magnetic data augmentation task, which are trained using the Stochastic Gradient algorithm on a cluster server with Intel(R) Xeon(R) CPU E5-2670 v3 for 60 epochs. The learning rate is initially set to 10^{-4}, and then reduced to 10^{-5} at 40 training epochs. The trained networks take approximately 7 seconds to segment 300 $64 \times 64 \times 6$-sized images. It can be seen from Table 2 that Our trained iCNN model achieves the segmentation accuracy of 93.98% on the test dataset of 300 patches and its Resnet counterpart achieves the accuracy of 93.23%. Our iCNN model adds the trivial cost of the implicit steps [15], however, it performs slightly better in terms of segmentation accuracy and exhibits faster convergence of the training than its ResNet counterpart on the magnetic dataset (see Figure 10)

Network	approximate training time	Accuracy
iCNN	420 min	93.98%
ResNet Counterpart	400 min	93.23%

Table 2: Comparison of iCNN and its ResNet counterpart on the magnetic dataset

Figure 9 shows automatic segmentation of 16 out of 300 test patches with our iCNN model, depicted alongside the manual segmentation. The automatic segmentation of these test patches is somewhat different from the manual segmentation. Since only one single geological map is available for our magnetic data, it is not clear whether these differences lie in the range of the deviations

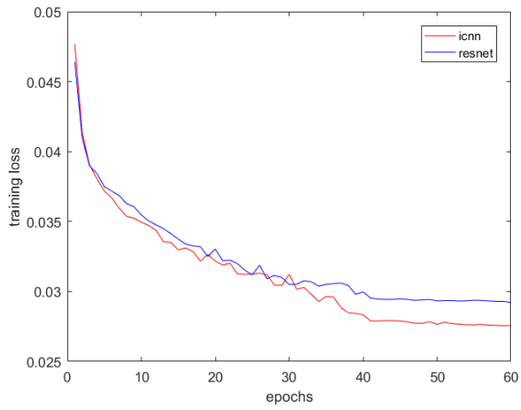


Figure 10: Convergence of the training using iCNN and its ResNet counterpart on the magnetic dataset.

between experts. Automatic segmentation of magnetic data which are independently labeled by different experts may be an item of future work. Despite these differences, the segmentation accuracy implies sufficient overlap with manual segmentation. In addition, as soon as the training stage is finished, in our case, only around 7 seconds are needed to segment 300 patches of the size $64 \times 64 \times 6$, which is impossible for an experienced geologist. These observations suggest that we could potentially replace human segmentation of magnetic data with our method, or at least take the automatic segmentation results as a starting point to assist the experts with their manual segmentation.

7 Conclusion and outlook

7.1 Summary

In this work we have discussed the problem of semantic segmentation of magnetic data. We have addressed the issues data data pre-processing and data augmentation as well as used a deep neural network architecture that works well for the problem at hand.

While data pre-processing of magnetic data is relatively straight forward, special attention was given to the data augmentation using MPS. To our knowledge, this is the first time that such geostatistical technique is used in the context of network training. Using this technique enables us to overcome the extremely insufficient (only one in our case) annotated magnetic data. The results show that we can generate sufficient number of realizations which are geologically realistic and preserve the complex geometry of magnetic sources. One has to always remember that the MPS samples from a distribution that is similar to the image at hand. Therefore, the trained AI system is expected to work well on problems where the geology is similar to the training image. Care must be taken to either not use, or add examples in areas where the geostatistics is different than the one trained on.

As for a network architecture, we also propose the iCNN to automatically segment magnetic data. The method has properties similar to U-net and, as explored in the literature, it is typically better than a simple ResNet. The result suggests that our method is comparable to human seg-

mentation, or at least can be used as a springboard to aid in manual segmentation. To the best of our knowledge, we are not aware of any work, at least in machine learning community, that tries to automate segmentation of magnetic data.

Acknowledgments

The funding for this work is provided through the University of British Columbia's Four-Year-Fellowship program.

Computer code availability

Code name	magSemanticSegAlgorithm
Developers	Xiaojin Tan and Eldad Haber
Contact address	2020-2207 Main Mall, University of British Columbia, Canada
Phone number	+01 778-386-0768
Email address	xtan@eoas.ubc.ca
Hardware required	Intel Xeon E5-2670 V3 Dodeca-core 2.30 Ghz Processor; 128 GB RAM
Software required	MATLAB R2016b or newer version
Program language	MATLAB
Program size	30M
link to the code	`https://github.com/tanxiaojin/magSemanticSegAlgorithm.git`

Table 3: Information about the computer code

The name of code is "magSemanticSegAlgorithm", which stands for the algorithm of magnetic semantic segmentation. The relevant information about the code is shown in Table 3.

References

[1] G Burc Arpat and Jef Caers. Conditional simulation with patterns. *Mathematical Geology*, 39(2):177–203, 2007.

[2] Uri M Ascher, Steven J Ruuth, and Brian TR Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.

[3] F Gregory Ashby and Daniel M Ennis. Similarity measures. *Scholarpedia*, 2(12):4116, 2007.

[4] 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.

[5] R. Blakely. *Potential theory in gravity and magnetic applications*. Cambridge university press, 1996.

[6] Peter J Burt and Edward H Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)*, 2(4):217–236, 1983.

[7] Alexander Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I Iglovikov, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *arXiv preprint arXiv:1809.06839*, 2018.

[8] Cleyton de Carvalho Carneiro, Stephen James Fraser, Alvaro Penteado Crósta, Adalene Moreira Silva, and Carlos Eduardo de Mesquita Barros. Semiautomated geologic mapping using self-organizing maps and airborne geophysics in the brazilian amazon. *Geophysics*, 77(4):K17–K24, 2012.

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

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

[12] Ayoub El Ouassini, Antoine Saucier, Denis Marcotte, and Basil D Favis. A patchwork approach to stochastic simulation: a route towards the analysis of morphology in multiphase systems. *Chaos, Solitons & Fractals*, 36(2):418–436, 2008.

[13] DF Graham and GF Bonham-Carter. Airborne radiometric data-a tool for reconnaissance geological mapping using a gis. *Photogrammetric Engineering and Remote Sensing*, 59(8):1243–1249, 1993.

[14] Felipe B Guardiano and R Mohan Srivastava. Multivariate geostatistics: beyond bivariate moments. In *Geostatistics Troia'92*, pages 133–144. Springer, 1993.

[15] Eldad Haber, Keegan Lensink, Eran Triester, and Lars Ruthotto. Imexnet: A forward stable deep neural network. *arXiv preprint arXiv:1903.02639*, 2019.

[16] Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *arXiv preprint arXiv:1705.03341*, 2017.

[17] Per Christian Hansen, James G Nagy, and Dianne P O'leary. *Deblurring images: matrices, spectra, and filtering*. SIAM, 2006.

[18] JR Harris. Clustering of gamma ray spectrometer data using a computer image analysis system. *Statistical Applications in the Earth Science. Geological Survey of Canada, Canada*, pages 19–31, 1989.

[19] JR Harris and Eric C Grunsky. Predictive lithological mapping of canada's north using random forest classification applied to geophysical and geochemical data. *Computers & Geosciences*, 80:9–25, 2015.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[21] Mehrdad Honarkhah and Jef Caers. Stochastic simulation of patterns using distance-based pattern modeling. *Mathematical Geosciences*, 42(5):487–517, 2010.

[22] Mehrdad Honarkhah and Jef Caers. Direct pattern-based simulation of non-stationary geostatistical models. *Mathematical geosciences*, 44(6):651–672, 2012.

[23] LY Hu and T Chugunova. Multiple-point geostatistics for modeling subsurface heterogeneity: A comprehensive review. *Water Resources Research*, 44(11), 2008.

[24] Vladimir Iglovikov, Sergey Mushinskiy, and Vladimir Osin. Satellite imagery feature detection using deep convolutional neural network: A kaggle competition. *arXiv preprint arXiv:1706.06169*, 2017.

[25] Baris Kayalibay, Grady Jensen, and Patrick van der Smagt. Cnn-based segmentation of medical imaging data. *arXiv preprint arXiv:1701.03056*, 2017.

[26] RH Keldermann, MP Nash, and AV Panfilov. Modeling cardiac mechano-electrical feedback using reaction-diffusion-mechanics systems. *Physica D: Nonlinear Phenomena*, 238(11-12):1000–1007, 2009.

[27] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[29] Stephen Kuhn, Matthew J Cracknell, and Anya M Reading. Lithologic mapping using random forests applied to geophysical and remote-sensing data: A demonstration study from the eastern goldfields of australia. *Geophysics*, 83(4):B183–B193, 2018.

[30] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] S Liu and T Mackey. Using images in a geological interpretation of magnetic data. In *AGSO Research Newsletter*, volume 28, pages 17–19, 1998.

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[33] Gregoire Mariethoz and Jef Caers. *Multiple-point geostatistics: stochastic modeling with training images*. John Wiley & Sons, 2014.

[34] Gregoire Mariethoz and Sylvain Lefebvre. Bridges between multiple-point geostatistics and texture synthesis: Review and guidelines for future research. *Computers & Geosciences*, 66:66–80, 2014.

[35] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[36] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[37] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. *arXiv preprint arXiv:1703.02719*, 2017.

[38] Augusto César Bittencourt Pires. Statistical characterization of greenstone sequences using magnetic and gamma-ray spectrometric data, guarinos-pilar de goiás area, brazil. *Brazilian Journal of Geology*, 20(1):293–298, 1990.

[39] Kris Popat and Rosalind W Picard. Novel cluster-based probability model for texture synthesis, classification, and compression. In *Visual Communications and Image Processing'93*, volume 2094, pages 756–769. International Society for Optics and Photonics, 1993.

[40] Walter R. Roest, Jacob Verhoef, and Mark Pilkington. Magnetic interpretation using the 3-d analytic signal. *Geophysics*, 22:4:591–592, 1991.

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[42] Selim S Seferbekov, Vladimir Iglovikov, Alexander Buslaev, and Alexey Shvets. Feature pyramid network for multi-class land segmentation. In *CVPR Workshops*, pages 272–275, 2018.

[43] Toshio Sekimura, Anotida Madzvamuse, Andrew J Wathen, and Philip K Maini. A model for colour pattern formation in the butterfly wing of papilio dardanus. *Proceedings of the Royal Society of London B: Biological Sciences*, 267(1446):851–859, 2000.

[44] Kenji Suzuki. Overview of deep learning in medical imaging. *Radiological physics and technology*, 10(3):257–273, 2017.

[45] Pejman Tahmasebi, Ardeshir Hezarkhani, and Muhammad Sahimi. Multiple-point geostatistical modeling based on the cross-correlation functions. *Computational Geosciences*, 16(3):779–797, 2012.

[46] William Murray Telford, WM Telford, LP Geldart, Robert E Sheriff, and Robert E Sheriff. *Applied geophysics*, volume 1. Cambridge university press, 1990.

[47] Ulrich Trottenberg, Cornelius W Oosterlee, and Anton Schuller. *Multigrid*. Academic press, 2000.

[48] Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *ACM SIGGRAPH Computer Graphics*, volume 25, pages 289–298. ACM, 1991.

[49] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016.

[50] Han Wang, Yongsheng Han, and Jinghai Li. Dominant role of compromise between diffusion and reaction in the formation of snow-shaped vaterite. *Crystal Growth & Design*, 13(5):1820–1825, 2013.

[51] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*, pages 93–117. Eurographics Association, 2009.

[52] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000.

[53] Andrew Witkin and Michael Kass. Reaction-diffusion textures. *ACM Siggraph Computer Graphics*, 25(4):299–308, 1991.

[54] Larry S Yaeger, Richard F Lyon, and Brandyn J Webb. Effective training of a neural network character classifier for word recognition. In *Advances in neural information processing systems*, pages 807–816, 1997.

[55] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[56] Tuanfeng Zhang, Paul Switzer, and Andre Journel. Filter-based classification of training image patterns for spatial simulation. *Mathematical Geology*, 38(1):63–80, 2006.

[57] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.