

## Research paper

# A convolutional neural network for semi-automated lineament detection and vectorisation of remote sensing data using probabilistic clustering: A method and a challenge



Amin Aghaee<sup>a,\*</sup>, Pejman Shamsipour<sup>b</sup>, Shawn Hood<sup>a</sup>, Rasmus Haugaard<sup>c</sup>

<sup>a</sup> GoldSpot Discoveries Corp., 69 Yonge Street, Suite 1010, M5E1K3, Toronto, Ontario, Canada

<sup>b</sup> GoldSpot Discoveries Corp., Technical Office, Montreal, Quebec 980 Rue Cherrier, Suite 201H2L 1H7, Canada

<sup>c</sup> Metal Earth, Mineral Exploration Research Centre, Harquail School of Earth Sciences, Laurentian University, Sudbury, P3E2C6, Ontario, Canada

## ARTICLE INFO

**Keywords:**

Convolutional neural networks  
Lineaments  
Geology  
Clustering  
Geophysics

## ABSTRACT

In this paper we present the framework and open-source software code to train and apply Deep Learning Convolutional Neural Networks (CNNs) to the prediction of geological lineaments using topographic, magnetic, and gravity raster data. Many important applications relate to the recognition of linear geological structures from remote sensing data, such as thrust faults, bedrock fault and shear zones, lithological contacts, fractures and fold structures. The digitization of fault lineaments is conventionally performed by geologists or geophysicists with working knowledge of the relevant data, e.g., topographic Digital Elevation Model, magnetic data, and gravity data. Visual inspection and extraction is simple but subjective; the process is also time-expensive with efficiency and accuracy depending on the individual's knowledge, experience, and skill. For decades there has been interest in ways to automate this process. Our CNN approach is trained using publicly available lineament GIS data from the Quest BC project in British Columbia, Canada, and the Loch Lilly-Kars area of New South Wales, Australia. The datasets used to train the prediction models resulted in interesting predictions proximal to the training areas: some major lineaments are indicated, some are missed, and potential new (valid) lineaments are indicated. The results indicate potential for use as a semi-automated lineament detection solution. In contrast, but as anticipated, application of the model to the blind-test area of the Swayze greenstone belt, Ontario, produced poor lineament prediction results (as compared to publicly available interpretations). We interpret this result as related to insufficiently large training data inputs. However, it is inferred that results could be improved through feature engineering (e.g., use of topographic slope, rather than simply elevation) without the need to simply create larger training datasets. We hope that, by making the code open-source, the geoscience community will use this platform to gradually improve an open source fault prediction model.

## 1. Introduction

The recognition of geological linear structures in bedrocks such as thrust fault and shear zones, lithological contacts, fractures and fold structures from remote sensing data has many important applications including ground water exploration (Sander, 2007; Chandra et al., 2006), assessment of nuclear storage locations (Middleton et al., 2015; Tiren, 2010), landslide and earthquake hazards (Ramli et al., 2010; Geiß and Taubenböck, 2013), and mineral resource exploration (Masoud and Koike, 2017; Liu et al., 2013). The location of a lineament can be identified by change in gradient magnitude, change in pattern, occurrence of linear minima/maxima, and displacement of interpreted or

known reference structures (Hobbs, 1904; Kocal et al., 2004; Vassilas et al., 2002; Ramli et al., 2010).

Lineament extraction from potential-field remote sensing data can be made visually, by hand, or automatically using software algorithms (Middleton et al., 2015). Automated, computer-based extraction of lineaments from remote sensing data contrasts with traditional methods which rely on the experience of experts. Visual inspection and extraction is simple, but subjective; the process is also time-expensive with efficiency and accuracy depending on the individual's knowledge, experience, and skill. In contrast, digital analysis and automated extraction consumes less time and so is more efficient than manual approaches. The process is objective, but can be computationally complex and

\* Corresponding author.

E-mail addresses: [amin@goldspot.ca](mailto:amin@goldspot.ca) (A. Aghaee), [pejman@goldspot.ca](mailto:pejman@goldspot.ca) (P. Shamsipour), [shawn@goldspot.ca](mailto:shawn@goldspot.ca) (S. Hood), [RNielsen@laurentian.ca](mailto:RNielsen@laurentian.ca) (R. Haugaard).

susceptible to edge-cases. False positives and false negatives are possible with both manual and automated techniques. However, automated techniques (such as those applying artificial intelligence) provide the additional benefit of handling large datasets produced from varied and numerous available remote sensing data sources (Ma et al., 2015).

Many reviews summarize the manual and automatic techniques for extracting lineament data and producing interpreted map (sometimes also referred to as a thematic map) of geological structural data from remote sensing information (Ray, 1960; Paine and Kiser, 2012; Lillesand et al., 2015; Ran et al., 2017; Tiren, 2010; Ramli et al., 2010; Isles and Rankin, 2013). No single-edge detector method is a perfect solution for geologic mapping, and combined approaches tend to produce highest confidence results for a given contact location (Cooper and Cowan, 2006; Pilkington and Keating, 2004). For this reason, machine learning and deep learning (artificial intelligence sub-disciplines) provide valuable approaches for lineament detection Lary et al. (2016). Deep learning Convolutional Neural Networks (CNNs) Krizhevsky et al. (2012) offer an opportunity to leverage manual expert interpretation of geophysical datasets in rasterized form (i.e., as training data) to create robust feature detection models of the gridded remote sensing images. These in turn can be interpreted by geological experts, who are trained to judge their utility and accuracy. A problem with CNNs is that they can be difficult to tune, may not generalise well to new areas, have decreasing performance as image complexity increases, and can be subject to “model drift” (where accuracy and precision degrade over time, as input data shifts in character) (Tudor Ionescu et al., 2016; Spanhol et al., 2016; Xu et al., 2019; Xie et al., 2016; Srivastava et al., 2014). However, some of these issues relate to the typical use of CNNs to classify imagery of objects. In contrast, lineament analysis from remote sensing data is predictable enough that easy-to-learn workflows are taught on the subject because the patterns involved are generally consistent across the Earth's crust Isles and Rankin (2013).

The CNN code for this project is provided on Github, along with a visualiser for data in the form of an applet <https://github.com/aminrd/LineamentLearning>. The intention is to provide a common open-source option for those who work with remote sensing geophysics data, such that the wider community can tackle the generalisation problem presented in this paper.

## 2. Background

Given a dataset of remote sensing data, the problem of lineament detection is typically to find lineaments within the represented region. In the present study, we look to create a Deep Learning model trained on two particular regions, and investigate model performance when applied to other geographical areas. Feature input data includes magnetic maps, gravity maps and digital elevation maps. Train/test data are represented by maps of lineaments interpreted by geoscience experts, available from respective national or state geological surveys. These training data are considered as representing the model's desired output. We divide input images and interpreted image into a training set and a test set. We train our model using the training set. The resulting Deep Learning model is then used to predict lineaments in other areas in the world, using similar data stacks. Model predictions are assessed using the test data set, as well as the case study regions, on the basis of statistical performance and the quality of interpreted lineaments.

A typical use of convolutional networks is to recognize visual patterns directly from pixel images and to predict a label for those images. For instance, labels could be bicycle, cat, dog, car, etc. or a combination of those labels. Szegedy et al. (2015) propose a deep convolutional neural network architecture to detect objects in images that achieves the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge, 2014 (Krause et al. (2015)). We apply this approach to lineament detection from remote sensing data. In our model, the desired output should include localization which is an assignation of a prediction (the probability that a lineament is close

the pixel) to each pixel of the image. We examine results for the test set derived from the regions of training data, which are geologically complex in terms of faulting and folding. We then compare the quality of results from these areas to another area, regionally separate from the training data but geologically comparable.

## 3. Data: geological study areas

Case study areas have been selected of areas studied by federal or state geological surveys. The areas chosen are of interest because: (1) they include a variety of geology, in terms of rock types, structure formations and ages, (2) data include public gridded raster GIS data including topography, magnetic and gravity data, which has been relatively recently acquired, (3) they represent areas of known mineral deposit interest, but where overlying cover material obscures and restricts interpretation of bedrock, and (4) GIS lineament interpretations are available, and have been made by geoscience experts based mainly on observations from available geophysical data.

### 3.1. Train and test area for deep learning model

We used two datasets, Loch Lilly-Kars Belt, New South Wales, Australia and the Quesnel Terrane QUEST area, British Columbia, Canada, that we trained our model on and kept part of it for testing. In this section, we have a quick review on these datasets.

#### 3.1.1. Loch Lilly-Kars belt, New South Wales, Australia

The Loch Lilly-Kars Belt includes an area of about 40 by 200 km trending northeast to southwest along the southern margin of the Paleoproterozoic Broken Hill Block of New South Wales, Australia (Baatar et al., 2020; Sharp, 2006). Although the belt is proximal to the historic Broken Hill mining camp, an exceptionally well-endowed lead-zinc region Woodward (1965); Greenfield et al. (2003), the Loch Lilly-Kars region is interpreted as an extension of the Mount Read Volcanic Belt in western Tasmania, which hosts several world class base and precious-metal deposits and related mines (Brathwaite, 1974; Green et al., 1981; Corbett, 1992; Baatar et al., 2020). Bedrock of the belt is completely overlain by tens to hundred meters of Mesozoic to Cenozoic aged sediments, meaning that understanding of the rocks beneath the cover has almost exclusively been interpreted using airborne geophysical data. The currently documented basement rocks include undated tholeiitic metabasalts and metadolerites, and some intermediate to felsic igneous rocks Baatar et al. (2020). Geophysical data and lineament interpretation for the area is curated by the Geological Survey of New South Wales Sharp (2006).

#### 3.1.2. The Quesnel Terrane QUEST area, British Columbia, Canada

In 2011, a program was launched by the Geoscience BC to stimulate exploration in the northwestern part of the province in the Quesnel Terrane. This area is interesting because it has good potential for copper and gold porphyry deposits.

Two airborne geophysical surveys was included in this project, Gravity and electromagnetic (EM) surveys. In this study, a collection and analysis of new regional stream sediment samples, a reanalysis of stream sediment samples and the new bedrock mapping is described as well. Geophysical data and lineament interpretation for the area is curated by Geoscience BC (Reichheld, 2012; Ltd., 2007b, a).

### 3.2. Geologically similar area, unrelated to the test data

We used dataset from the Swayze area of the well-known Neoproterozoic Abitibi Greenstone Belt (AGB) in northern Ontario, Canada, for testing the model's performance on an area that model has never been applied to before. However, this area is geologically similar to the Australian dataset we trained our model with. In this section, we have a quick review on the this dataset.

**Table 1**

List of input images for the lineament model.

| Layer 1 | Reduced to Pole (RTP) Map (magnetic data)             |
|---------|---|
| Layer 2 | 1st vertical derivative of RTP Map                    |
| Layer 3 | RTP map of high-resolution magnetic field of a region |
| Layer 4 | Residual shallow horizontal gradient for RTP Map      |
| Layer 5 | Residual intermediate horizontal gradient for RTP Map |
| Layer 6 | Residual deep horizontal gradient for RTP Map         |
| Layer 7 | Digital Elevation Map                                 |
| Layer 8 | Gravity Map   |

The Swayze area encompasses the western extent of the AGB. This area is attractive to many exploration geologists because of the main unsolved question in this area. The question is, why the area historically has been less mineral endowed than the rest of the AGB further to the east? This understanding shifted with the discovery of the > 1 Moz Au  $C^o t^-$  Gold Au–Cu deposit in 2009 [Rogers et al. \(2013\)](#) that in turn opened up new exploration projects in southern part of the Swayze area.

Swayze area contains a total of five banded iron formations that are intercalated with large volume E-W striking mafic and volcanic rocks ([Katz et al., 2017](#)). The volcanic rocks are often represented by a number

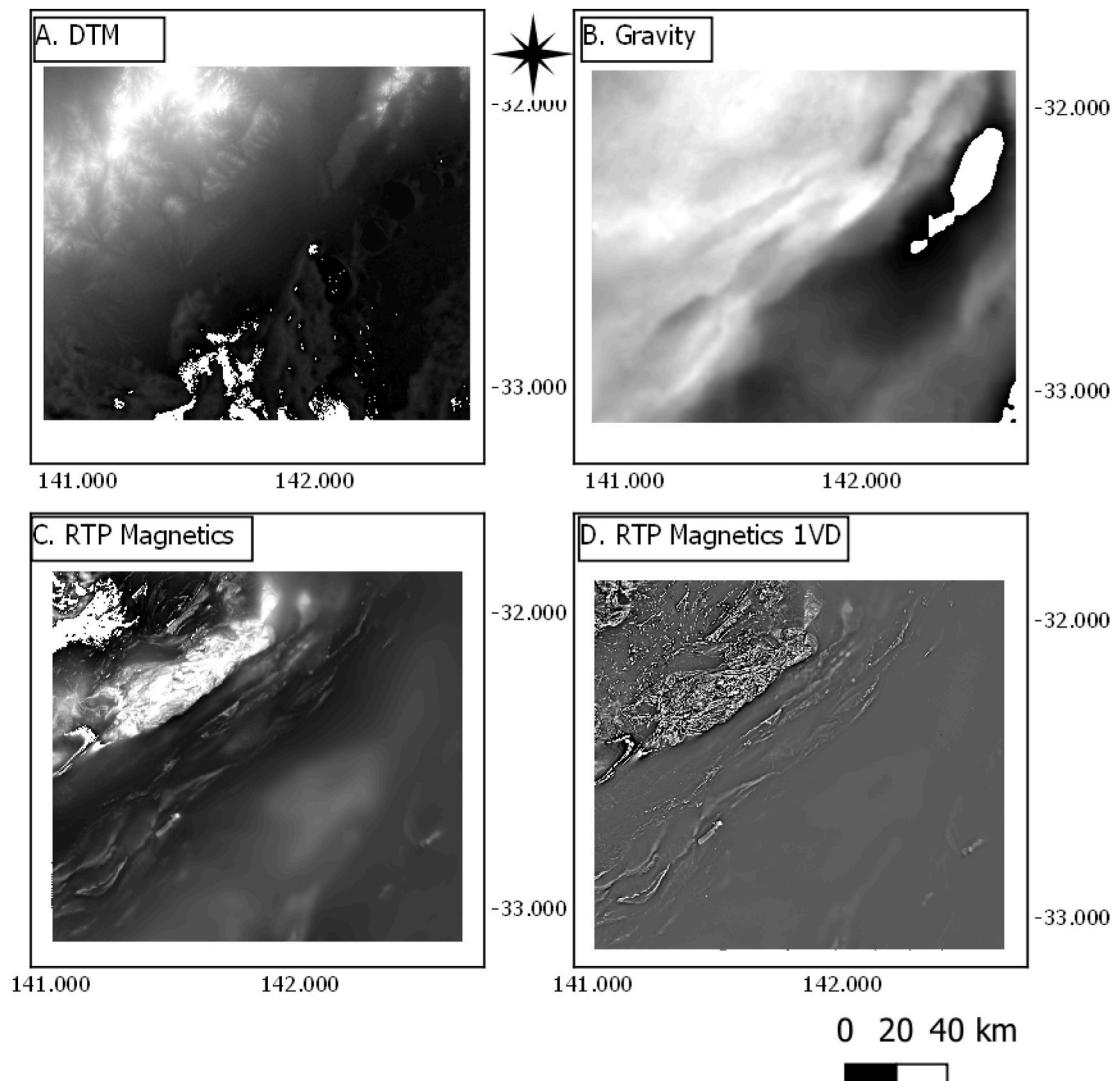
of fold-repetitions and are uncomfortably overlain by classic sedimentary rock. Three major E-W striking deformation zones run through the Swayze area. These deep linear crustal faults can potentially be linked to the rest of the AGB and therefore have gained interests from many exploration companies. [Heather et al. \(1995\)](#). 4r amin-here.

Generally, the area has a low degree of exposures. Multiple lithological contacts, however, exists and have been discovered by either outcrop mapping or the use of geochronology [Ayer et al. \(2002b, a\)](#). Essentially, the area is geometrically complex and exhibits a variety of geological lineaments. As a consequence, this area attracts many mining and exploration companies.

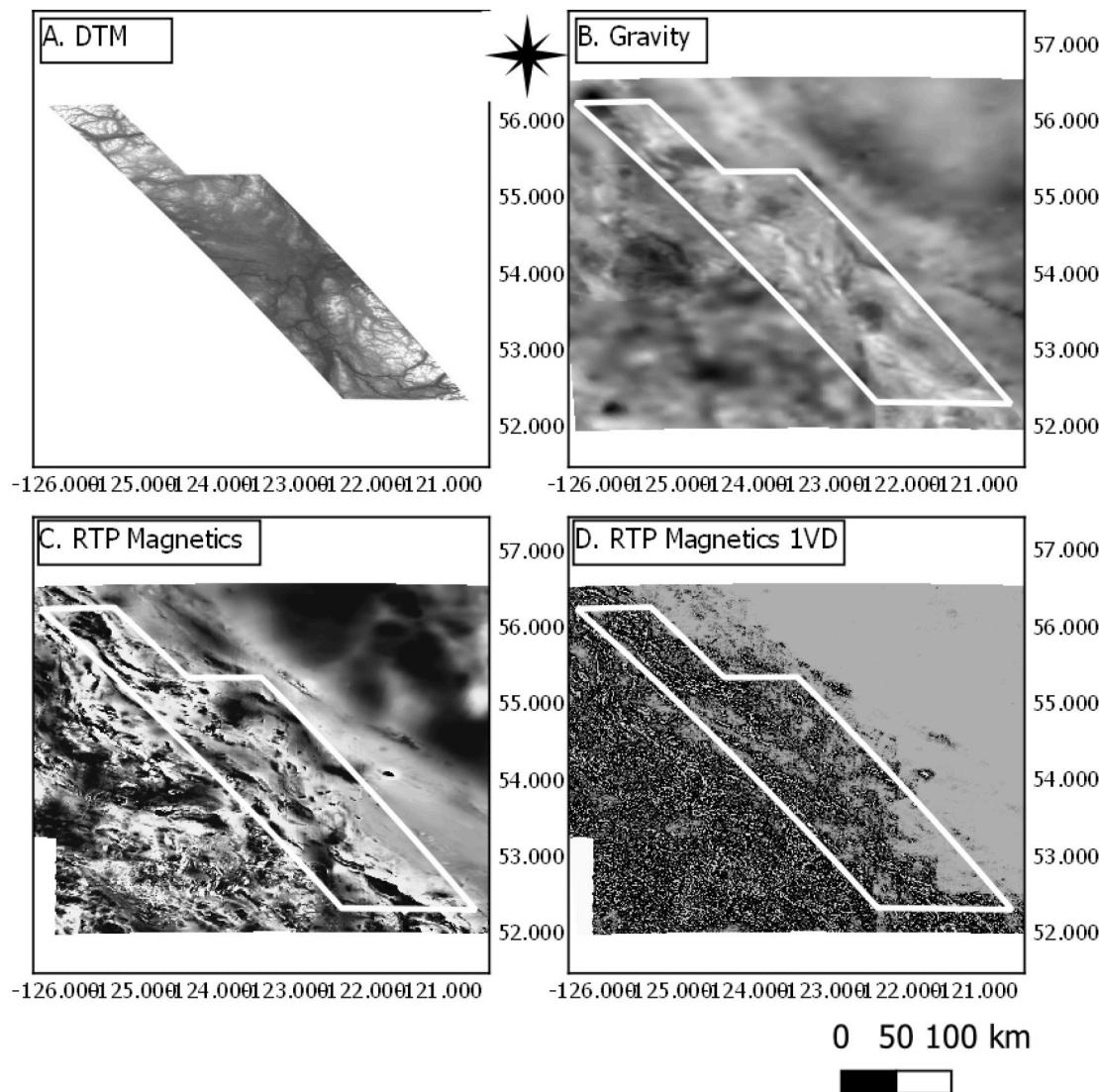
#### 4. Data preparation

**Table 1** lists all input layers used for model training.

Thousands of training images are typically required to properly train a neural network. This amount is difficult to achieve in geology (because acquiring data is expensive; input images are obtained from satellites or regional flying and the output map requires interpretation by experts). We generate many training samples out of the few input images we are given. As [Ciresan et al. \(2012\)](#) suggest, we train our model using a



**Fig. 1.** Australian study area geophysics input. A. Digital Terrain Model (DTM). B. Bouguer gravity. C. Reduced to pole magnetics. D. First vertical derivative of Reduced to pole magnetics.



**Fig. 2.** QUEST BC study area geophysics input. A. Digital Terrain Model (DTM). B. Bouguer gravity. C. Reduced to pole magnetics. D. First vertical derivative of Reduced to pole magnetics.

sliding-window to generate more training samples and to make our model translation invariant. Our model predicts a label (the probability that a lineament is close the pixel) for each pixel by providing a neighborhood of the pixel. We define these local regions as patches which are  $W \times W \times 8$  matrices where  $W$  is window size (because we have eight types of input images). (See Figs. 1-3)

In order to generate even more samples to train our model, and to make our model rotation-invariant (since aerial images could be captured from flying on any direction, and feature azimuth can occur in any orientation) we rotate every patch by  $r$  radians where  $r \in \text{Uniform}(0, 2\pi)$ . A target value or label for each patch derives from the center pixel of that patch. We want to have patches that are independent of rotations. So, we apply a circular mask filter to each patch as it is demonstrated in Fig. 4.

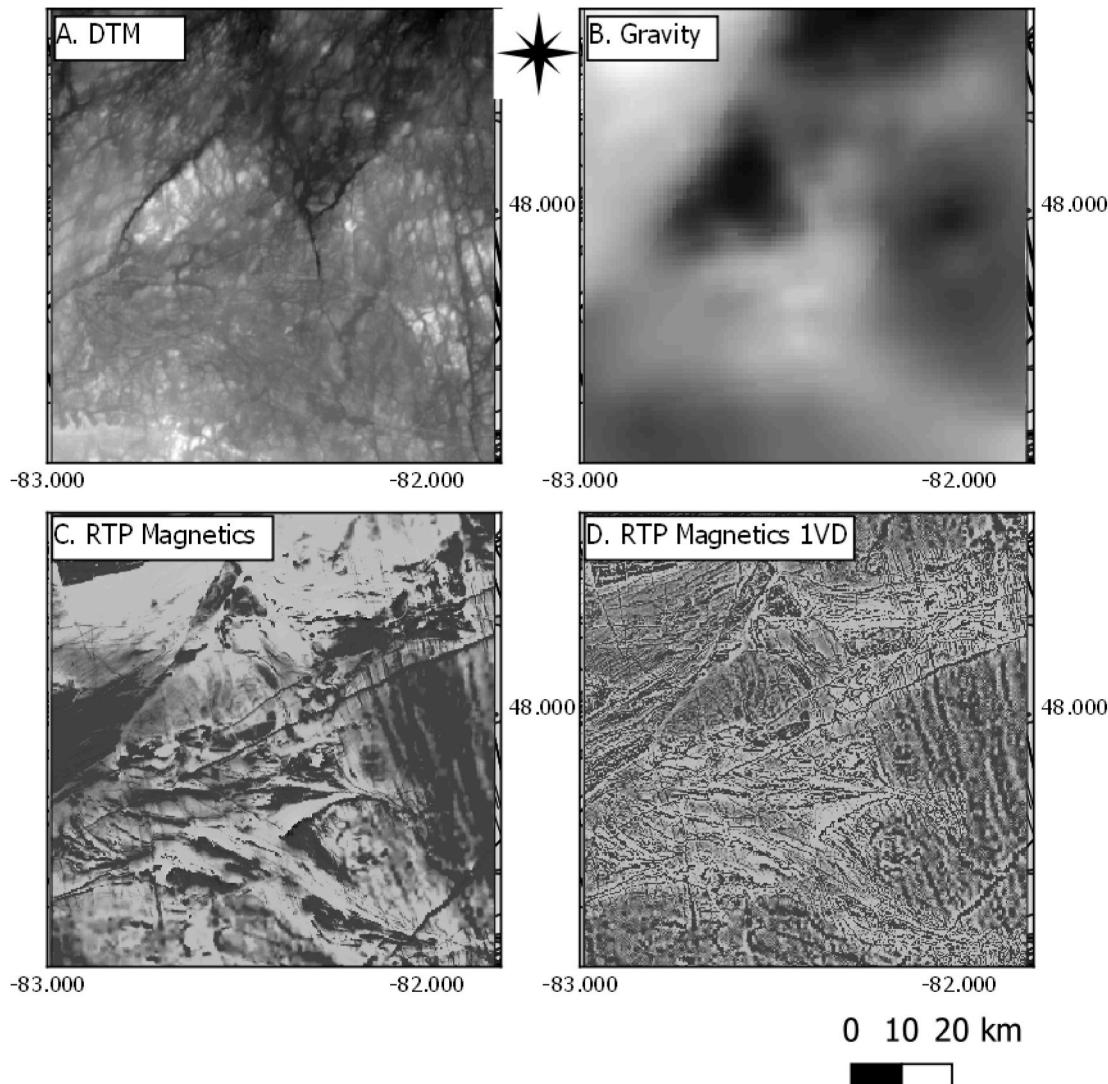
Each patch in the original dataset has a label  $y$  that corresponds to whether the center pixel of the patch has a drawn-by-expert lineament passing through it.

$p = 1$  A lineament drawn by expert passes through the center pixel  
 $p = 0$  No lineaments drawn by expert passes through the center pixel

Treating lineaments as 1-pixel wide lines does not work well for a number of reasons. Some areas in images could be noisy or could have not enough clue for experts to consistently draw a lineament. It is still acceptable to mineral explorers who use these interpretations if the model predicts the existence of lineaments with a few hundred meters accuracy. Accordingly, we expand interpreted lines using a probability distribution around the lines and convert the binary map into a probability map. The distribution should have a high probability at the center of the lines and low probabilities in farther pixels. We use a Gaussian distribution with  $\sigma, \mu$  parameters for expanding lineaments, with minimum value of  $\tau$  (values below this cutoff threshold will be zero). So, labels  $p$  are now a real number  $0 \leq p \leq 1$ . We have implemented this process inside a expansion function that has three parameters:

- The interpreted-by-expert map of lineaments
- The distribution function which is a Gaussian function  $G(\sigma = 1, \mu)$
- A cutoff threshold  $\tau$

After expanding lineaments using the Gaussian distribution, a sliding-window moves randomly on the map and generates the training samples. A workflow for preparing dataset with a given window size  $W$



**Fig. 3.** Swayze study area geophysics input. A. Digital Terrain Model (DTM). B. Bouguer gravity. C. Reduced to pole magnetics. D. First vertical derivative of Reduced to pole magnetics.

is described in [Algorithm 1](#). (See [Fig. 5](#))

#### Algorithm 1. Data Preparation Method

---

**Require:**  $W$  as window size  
**Require:**  $I$  as input images with size  $D_1 \times D_2 \times 8$   
**Require:**  $O$  as interpreted map

- 1: **while** Not Converged **do**
- 2:    $\tilde{E}_{D_1 \times D_2} \leftarrow \text{Expand}(O, G(\sigma = 1, \mu), \tau)$
- 3:    $(i, j) \leftarrow \text{choose randomly from } \left[ \frac{W}{2}, D_1 - \frac{W}{2} \right] \times \left[ \frac{W}{2}, D_2 - \frac{W}{2} \right]$
- 4:    $X \leftarrow \text{extract } W \times W \text{ window around } (i, j)$
- 5:    $X \leftarrow \text{rotate } X \text{ by } r \text{ radians where } r \in (0, 2\pi)$
- 6:    $X \leftarrow \text{apply a circular crop filter on } X$
- 7:    $y \leftarrow \tilde{E}(i, j)$
- 8:   Generate  $[X_{W \times W \times 8}, y]$
- 9: **end while**

---

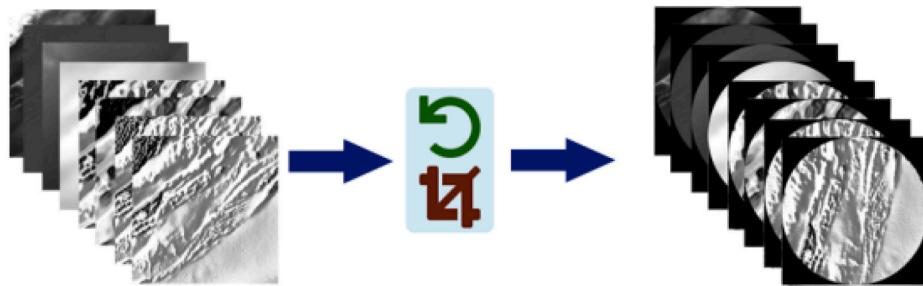
In this method the sliding window moves randomly on  $\left[ \frac{W}{2}, D_1 - \frac{W}{2} \right] \times \left[ \frac{W}{2}, D_2 - \frac{W}{2} \right]$  to avoid hitting the boundaries. At each iteration, this method generates a pair  $[X_{W \times W \times 8}, y]$ . The first element in that pair includes data from the processed input images and the second element is the desired output.

#### 4.1. Train convolutional neural network

During this work, we tested different neural networks with different structures from very simple models to more complicated ones like U-Net [Ronneberger et al. \(2015\)](#). Initially, we were trying networks with many convolutional layers that apply convolution operations to the inputs, passing results to the next layer. The convolution emulates the response of an individual neuron to visual stimuli. In our experiment, however, we realized that complex neural networks with many convolutional layers are not effective in this problem. This is the main reason why we decided to simplify our model.

The network architecture for windows size  $W = 49$  is illustrated in [Fig. 6](#). We take the input patches of size  $W \times W \times 8$  in the first layer (in this case  $w = 49$ ) and then convolve it with convolutional filters that are using  $3 \times 3$  convolution kernel with rectified linear unit (ReLU) applied to the convolutions' outputs. Then max pooling layers of size  $6 \times 6$  are used to reduce the size of the inputs and hence speed up the computation. After that, we use flatten operator which converts the previous layer into a single one-dimensional array. After flattening the previous layer, we create a fully-connected neural network with two hidden layers that use ReLU activation functions. Finally, our model has a single output with Sigmoid activation that generates the final prediction.

Since our output value is a probability value that corresponds to



**Fig. 4.** Rotation and Crop filters over every patch.

existence of lineament in the center pixel of input patch, we use cross-entropy or log-loss function for training. This loss function measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability or  $\hat{y}$  diverges from the actual label of  $y$ . The cost function is computed by taking the average of all cross-entropies in the sample. For example, suppose we have generated  $N$  samples  $(X_1, y_1), (X_2, y_2), \dots, (X_N, y_N)$ . The cross-entropy loss function is:

$$\text{Loss} = -\frac{1}{N} \sum_{n=1}^N \left[ y_n \cdot \log \hat{y}_n + (1 - y_n) \cdot \log (1 - \hat{y}_n) \right]$$

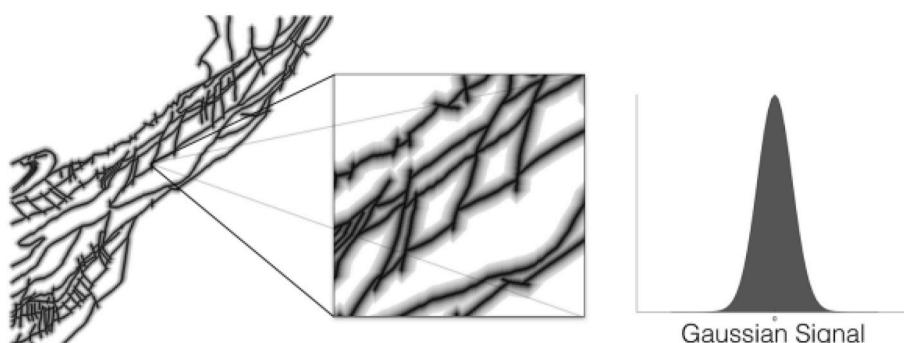
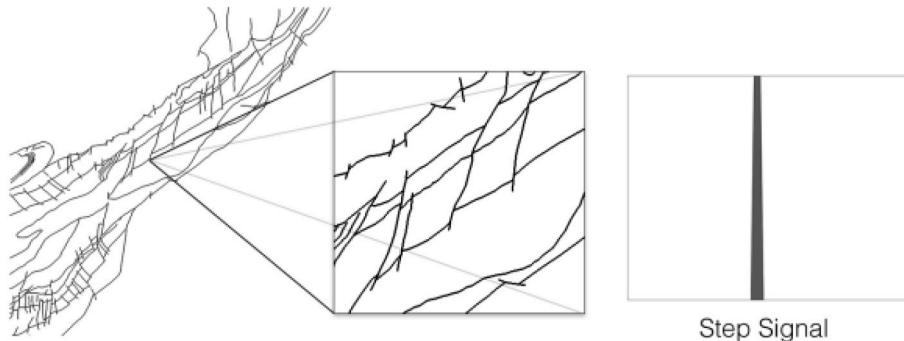
This function reaches its minimum when  $\hat{y}_n = y$ .

After training our convolutional neural network, we can make a probability map or Pmap for any region using our trained parameters. We predict values on the valid areas only. All non-valid areas such as

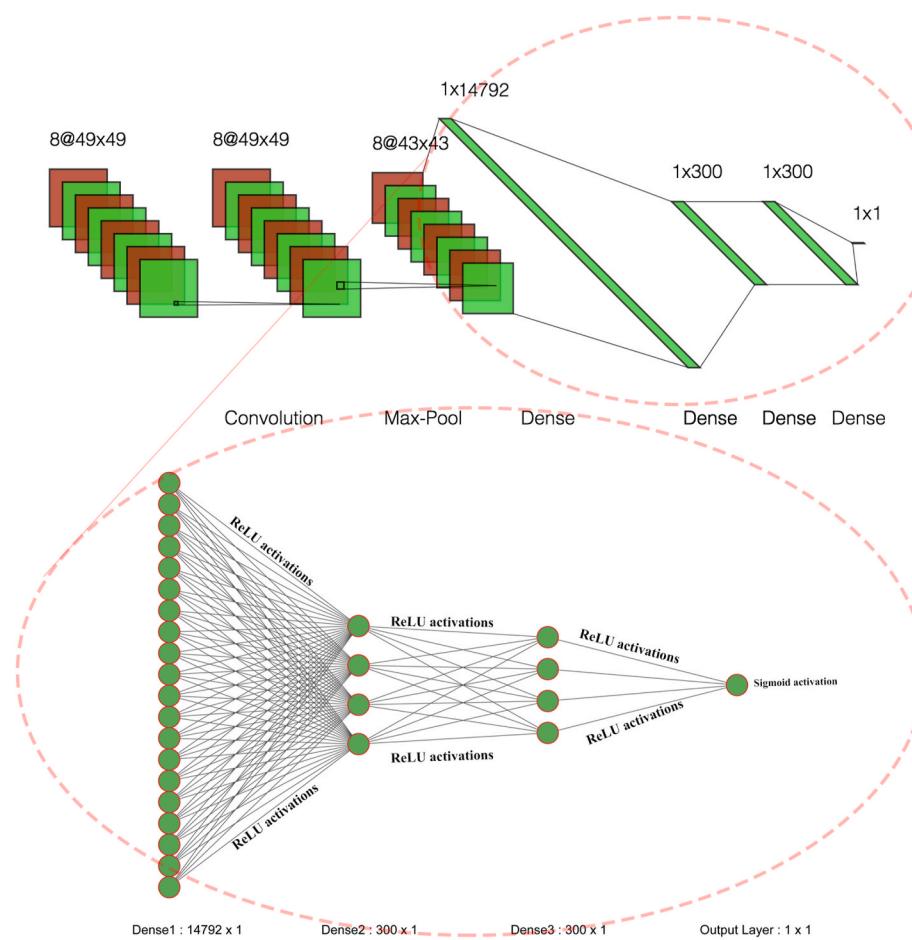
non-interpreted areas or out-of-boundaries are *NaN* (Not a Number). A Pmap is a two-dimensional map with size  $D_1 \times D_2$  (the same size as the input geological maps) where each pixel has a probability value between zero and 1. These values are our model's prediction on the probability of lineament existence on that pixel. A sample Pmap is illustrated in Fig. 7 where probabilities are visualized by shading of black where higher values are more black and lower values are more white. Non-valid areas with label *NaN* are also shown with white. We evaluate the output on valid areas only.

#### 4.2. Cluster analysis

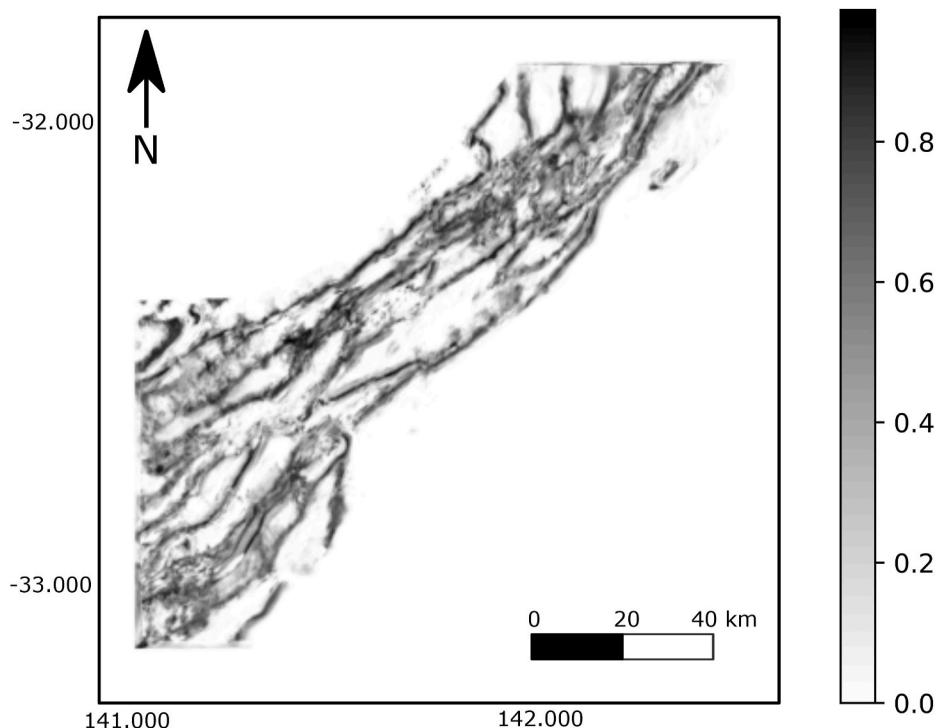
The expected final product of lineament model should be lines or curves. However, Pmaps are probability maps that have probability of fault existence in every location. In this section and the following



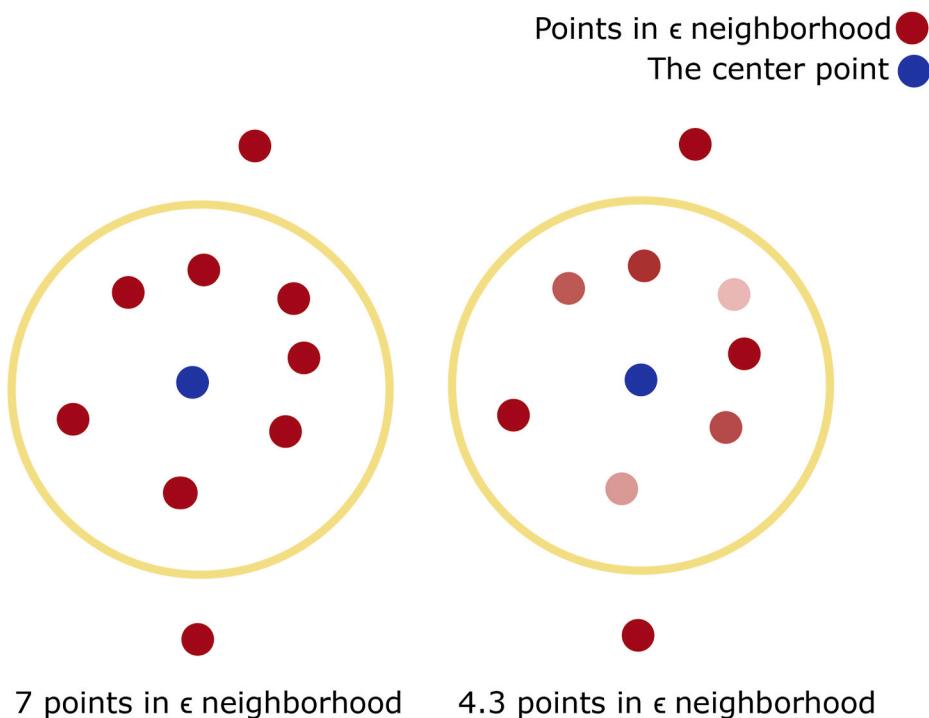
**Fig. 5.** Comparing expanded lineaments using Gaussian Signal with interpreted map.



**Fig. 6.** The convolutional neural network architecture for lineament learning.



**Fig. 7.** Probability map for the Australia data set with  $W = 45$ . Higher probabilities for a lineament are in black.



**Fig. 8.** Original DBSCAN versus P-DBSCAN. In P-DBSCAN, we sum up the probabilities of lineaments points in  $\epsilon$ -neighborhood. The image shown is for the Australian dataset.

section, we discuss our post-processing methods to convert Pmaps into lines and polynomials.

Density-based spatial clustering of applications with noise DBSCAN (Ester et al. (1996)) is a density-based clustering algorithm. Given a set of points in some space, it groups together points that are closely packed together (uses spatial distance as similarity function). It marks those points that lie alone in low-density regions as outliers (whose nearest neighbors are not close enough).

In the original DBSCAN method, a core point are points that have  $MinPts$  points in its  $\epsilon$ -neighborhood. We add a little modification into the original method. We call it P-DBSCAN or Probability-DBSCAN. In P-DBSCAN, lineament points have probabilities. So, we sum up the probabilities of lineaments points in  $\epsilon$ -neighborhood. Fig. 8 shows the  $\epsilon$ -neighborhood of the blue point in the original DBSCAN on the left and P-DBSCAN on the right.

Lineament points distributed on a Pmap are then passed to P-DBSCAN for clustering. A sample output with  $\epsilon = 3.3$  and  $MinPts = 20$  is illustrated in Fig. 9. In Fig. 9, each cluster is marked by a random color. P-DBSCAN, like DBSCAN, removes outliers by assigning no label to them. So, the nebulous and less confident areas in a Pmap are removed after running the clustering step.

#### 4.3. Curve fitting

In the final stage of our method, we convert output clusters of P-DBSCAN into curves and generate vector files as output. Each cluster initially consists of multiple points, with two-dimensional features represented by  $(x,y)$  values in 2D images. One option is to fit a linear regression model through each cluster and then replace that cluster with

the best model. However, this makes the output complex for the user looking at the results since there are many small curves. We use the following method to merge clusters so that the total number of curves get reduced:

#### Algorithm 2. Find optimum number of curves

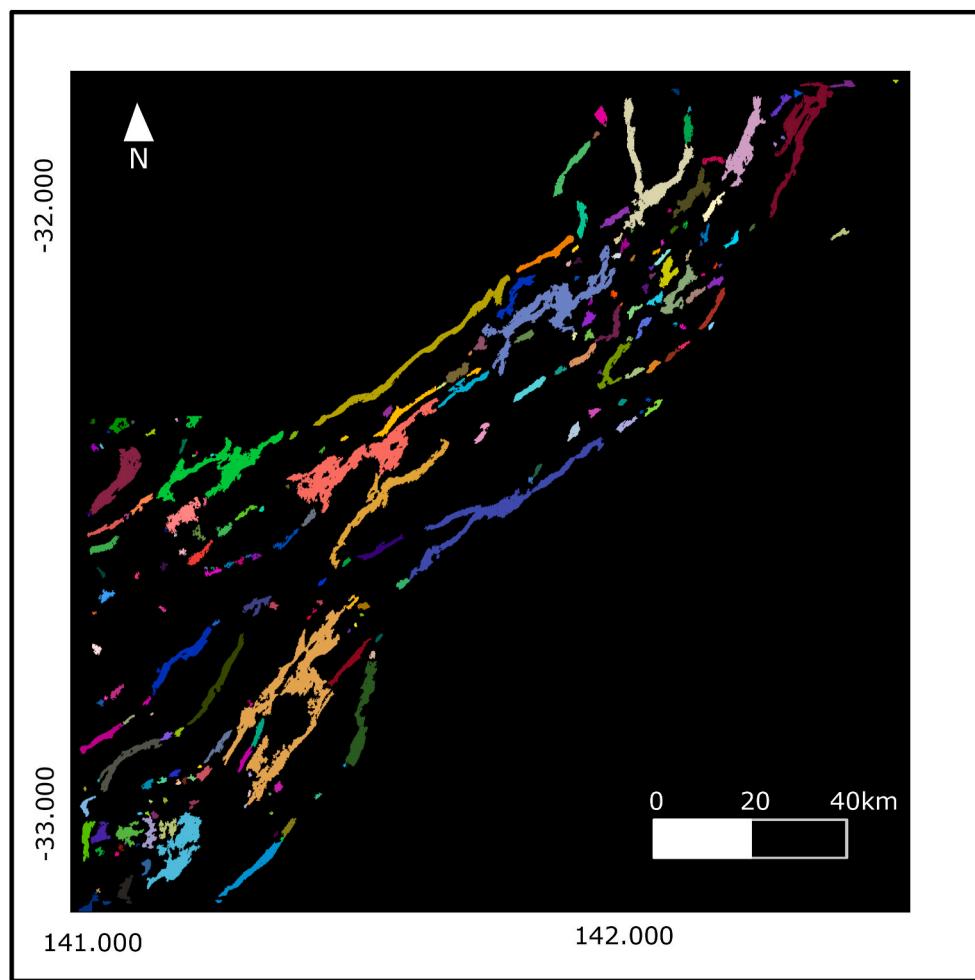
---

**Require:**  $C = c_1, c_2, \dots, c_k$  as clusters  
**Require:**  $M$  as maximum number of clusters  
**Require:**  $L$  as hyper-parameter for checking the nearest clusters  
**Require:**  $\lambda$  the error threshold

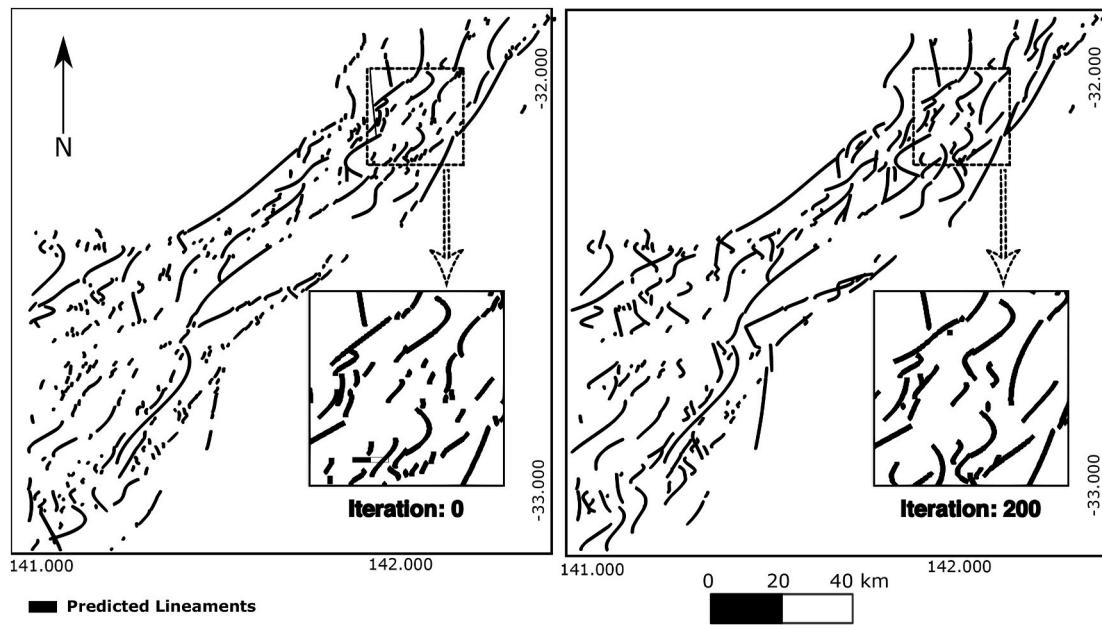
- 1: **while**  $\|C\| \geq M$  **do**
- 2:    $C_i \leftarrow$  choose the smallest cluster from  $C$
- 3:    $\{c'_1, c'_2, \dots, c'_L\} \leftarrow L$  closest clusters to  $C_i$
- 4:    $Best \leftarrow (c_i \cup c'_1)$
- 5:   **for**  $c_x \in \{c'_1, c'_2, \dots, c'_L\}$  **do**
- 6:     **if**  $Error(c_x \cup c_i), \lambda < Error(Best, \lambda)$  **then**
- 7:        $Best \leftarrow (c_i \cup c_x)$
- 8:     **end if**
- 9:   **end for**
- 10:   Merge  $Best$  and update  $C$
- 11: **end while**

---

In this method, function  $Error(cluster)$  returns mean square error of a best curve passing through a cluster. It fits linear, degree 2, degree 3 to degree 7 polynomials and choose the best curve based on our error function which relies on mean square error. Higher degree polynomials always have lower errors than lower degrees. However, we prefer simpler curves and simpler output. We choose low degree polynomials unless its error is greater than a threshold  $\lambda$ . We use the following method to find the error of fitting a polynomial:



**Fig. 9.** Running P-DBSCAN on Pmap with  $\varepsilon = 3.3$  and  $MinPts = 20$ . Each color represents a cluster in this map. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)



**Fig. 10.** Best Curves method iteration 0 and 200 on Australia dataset.

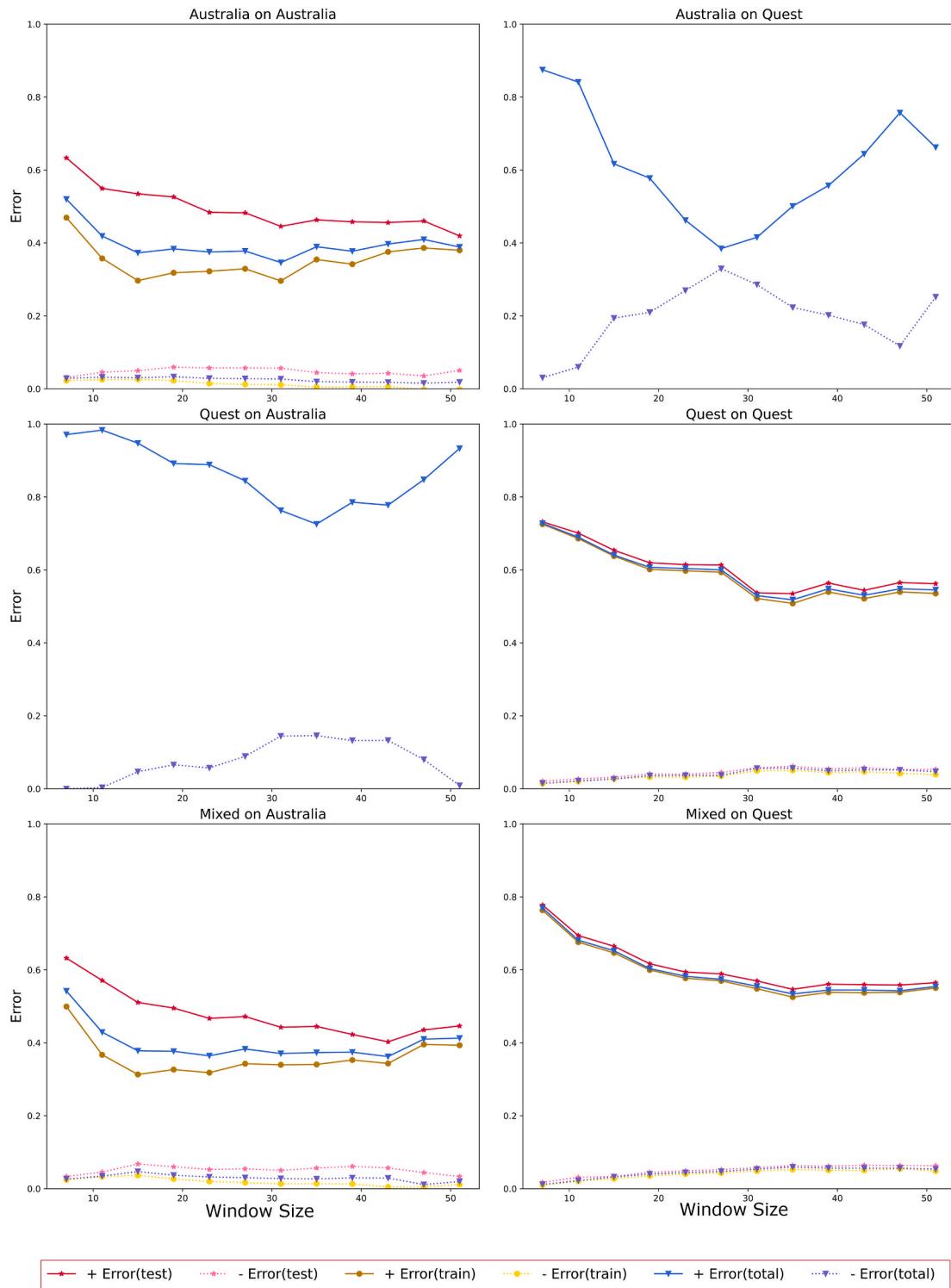
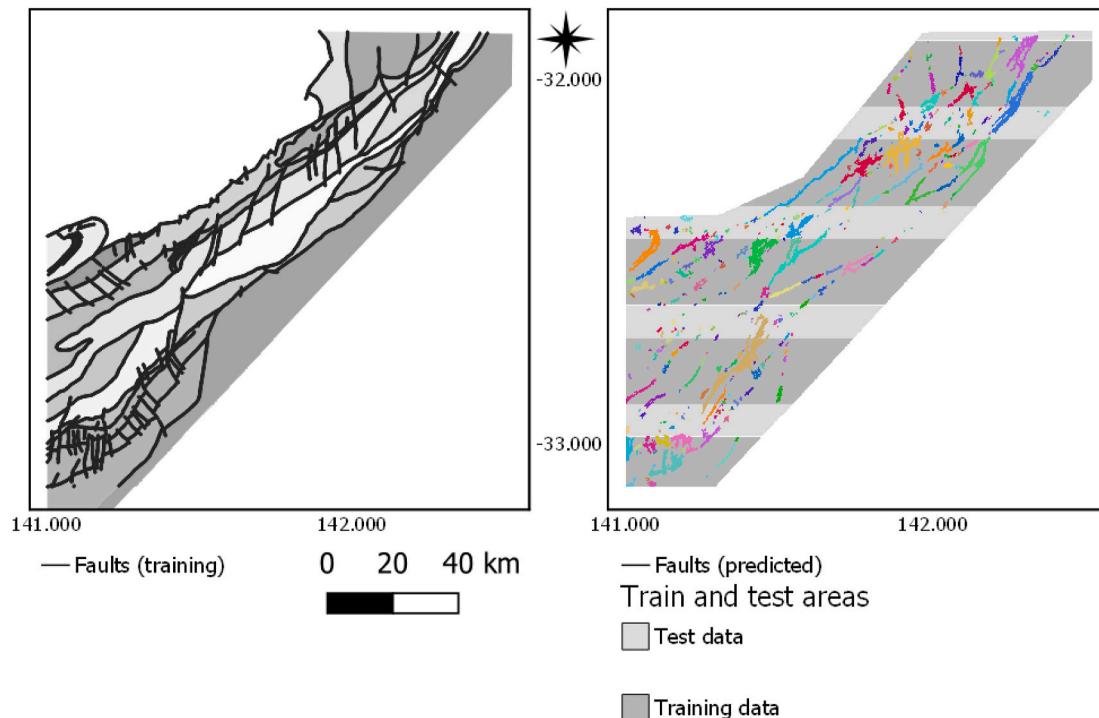


Fig. 11. Training, Testing and Total positive and negatives errors for different models.



**Fig. 12.** Model predictions results within the train and test areas (Loch Lilly-Kars, NSW, Australia) with true labels (lineaments from experts) on the left and model's lineament clustering on the right.

### Algorithm 3. Finding the error of fitting a polynomial

---

**Require:** Cluster  $C$   
**Require:** Threshold  $\lambda$

```

1: for  $d \in \{1, 2, \dots, 7\}$  do
2:   if  $Error(c_x \cup c_i), \lambda < Error(Best, \lambda)$  then
3:      $P \leftarrow$  fit degree  $d$  polynomial to  $C$ 
4:      $E \leftarrow$  mean square error for  $P$ 
5:     if  $E < \lambda$  then
6:       Return  $E$ 
7:     end if
8:   end if
9: end for
10: Return  $E$ 
```

---

Starting from  $K$  clusters, where  $k$  is the number of clusters initially found by clustering step, this method merges two clusters after each iteration and reduces the total number of clusters until it reaches  $M$ . The value of  $M$  is the desired number of clusters. On each iteration, it chooses the smallest cluster. We prefer removing smaller clusters and having larger clusters through which a single polynomial is fitted. Our method tests  $L$  nearby clusters one by one and chooses a cluster with a minimum curve error if those clusters are merged together. This method only tests  $L$  nearby clusters to merge nearby clusters in early steps. Then the clusters grow in size and get closer to each other.

The curves shown in 10 are outputs of our limited curve process initially and after 200 iterations. (See Fig. 10)

### 5. Model results on local area (train/test area)

We trained our model on two real datasets. Each dataset contains the input 8 layers listed in Table 1 and the interpreted fault lines. These datasets are:

- **Australia dataset:**  $106.5 \times 106.8$  kilometers or  $2130 \times 2136$  pixels (each pixel is 50 meters by 50 meters on the ground) from a relatively flat area in Australia.

- **Quest dataset:**  $253.9 \times 205.8$  kilometers or  $5078 \times 4116$  pixels (each pixel is 50 meters by 50 meters on the ground) from a mountainous region in center of British Columbia, Canada.

We used 66 percent of the area of these datasets for training purposes and 33 percent for testing and evaluations. In order to evaluate the outputs, we use the following notations:

- $p(x,y)$  is the desired output for the probability of lineament existence
- $\hat{p}(x,y)$  is the predicted value from the model
- $L^+ = \{(x,y) : p(x,y) > \tau\}$
- $L^- = \{(x,y) : p(x,y) < \tau\}$

We also define the positive error and the negative error as below:

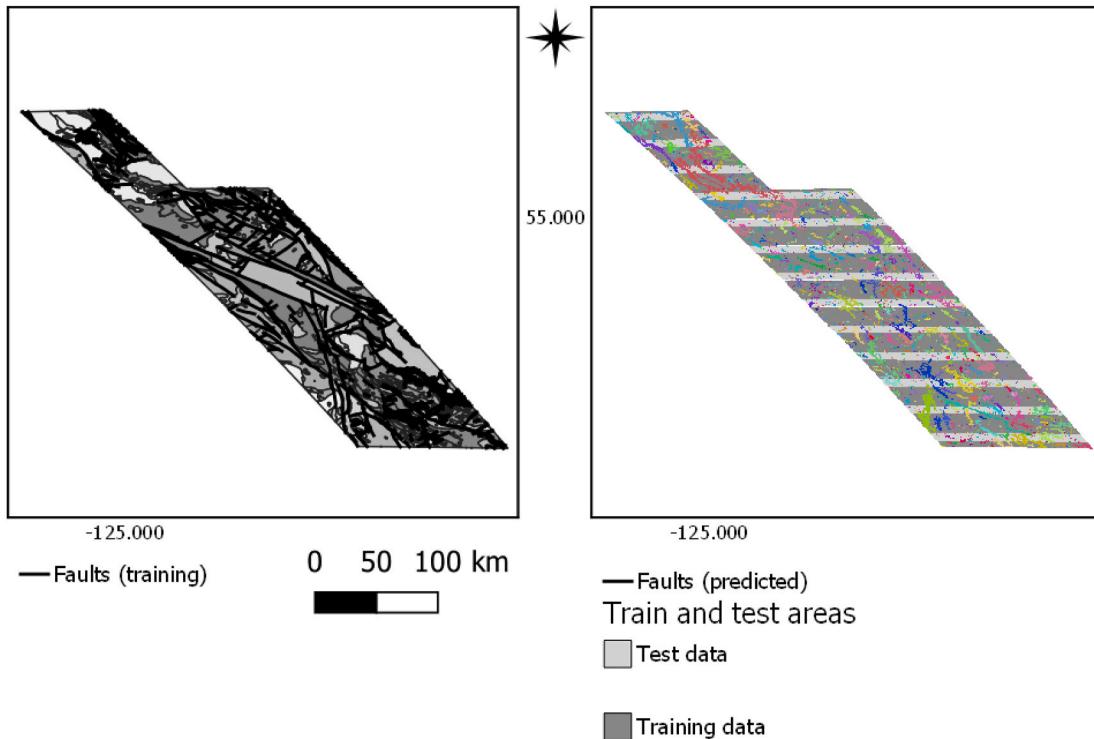
- Positive Error =  $\frac{1}{|L^+|} \sum_{(x,y) \in L^+} (p(x,y) - \hat{p}(x,y))^2$
- Negative Error =  $\frac{1}{|L^-|} \sum_{(x,y) \in L^-} \hat{p}(x,y)^2$

Thus, the positive error is mean square error of predictions on area where lineaments exist. The negative error is the mean square error of the predictions that are supposed to be zero. Consider an extreme predictor which always predicts  $\hat{p} = 1$ . The negative error computes the error on the regions that the desired output should be very low. Thus  $\hat{p} = 1$  predictor has a very high negative error. Another extreme predictor with  $\hat{p} = 0$  on the other hand, a negative error equal to zero.

In the training phase, three different types of models are trained:

- **Australia model:** These models are only trained on the Australia dataset
- **Quest model:** These models are only trained on the Quest dataset
- **Mixed model:** These models are trained on a mixed map of both the Quest and the Australia datasets

For data preparation, we use data preparation method discussed in Section 4. The mixed models are trained with a combination of Quest

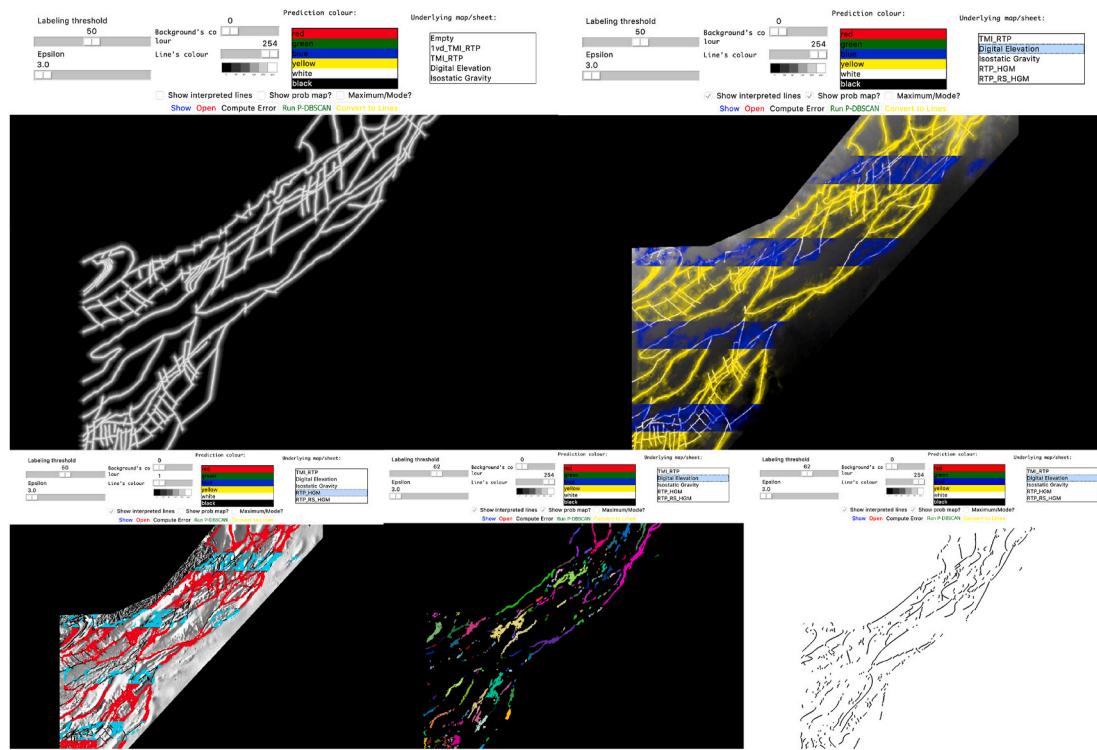


**Fig. 13.** Model predictions results within the train and test areas (Quest, BC, Canada.) with true labels (lineaments from experts) on the left and model's lineament clustering on the right.

and Australia datasets. That means we feed the mixed models with the data samples generated from the both datasets. For each type, we train various network structures with window size starting from  $W = 7$  pixels to  $W = 51$  pixels in increments of 4. Due to limitations on hardware memory and time limit, we did not train models with larger window

sizes. In Fig. 11 the positive errors and the negative errors for these variations are illustrated. The train, the test and the total errors are also included for these models.

By looking at the total errors in this figure, we see that they decrease by increasing the window size and converge or reaches to its minimum



**Fig. 14.** Small GIS applet screenshots.

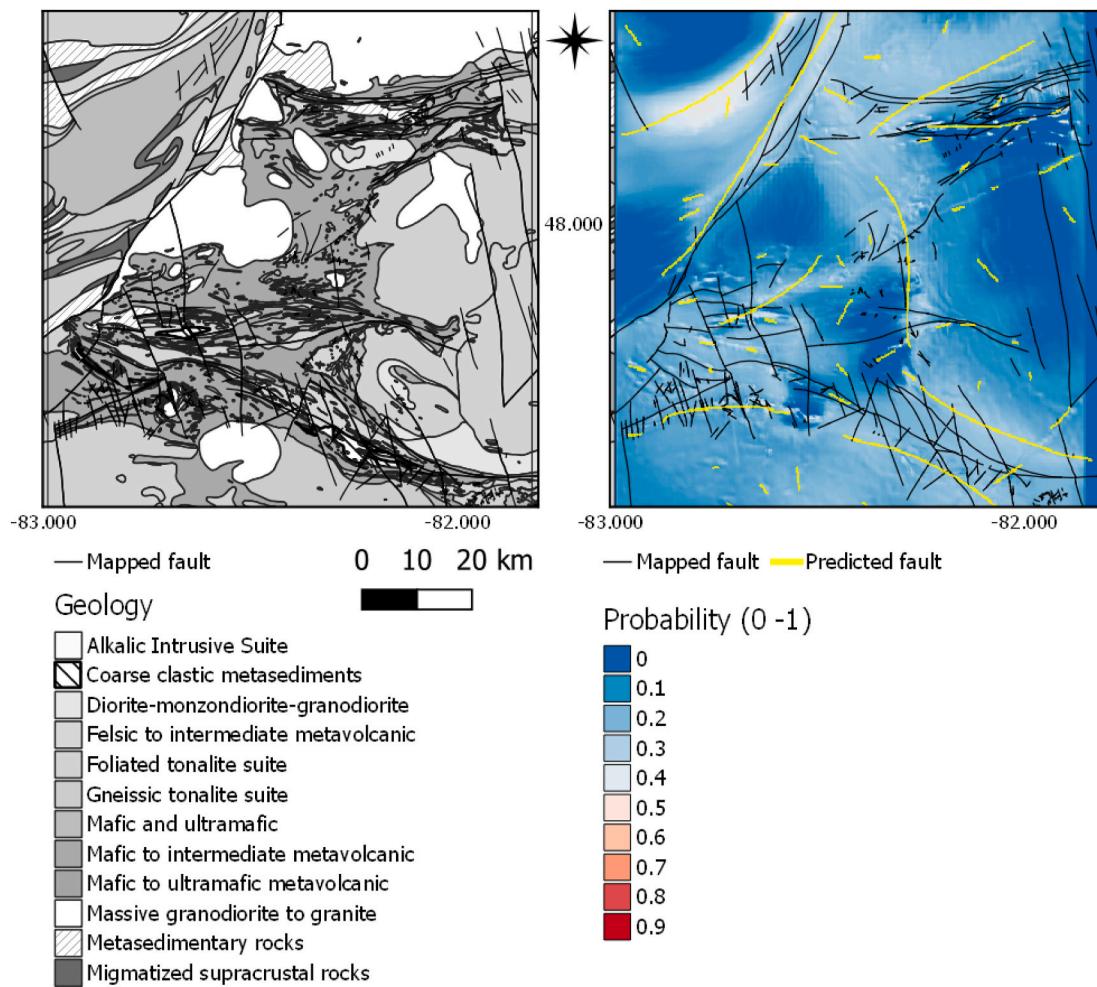


Fig. 15. Model predictions results of the Swayze area (geologically similar region).

approximately in range  $25 \leq W \leq 41$ . For example, Quest model on Quest dataset reaches to its minimum when  $w = 35$  but the Mixed model on Australia dataset reaches to its minimum when  $w = 23$ . Models that were trained on the Australia dataset, do not work well on the Quest dataset and vice versa. The Australia on Quest plot shows a high total negative error when total positive error is quite low. The Quest on Australia plot also shows a high total positive error. By looking at the Mixed on Australia and Mixed on Quest plots, we realize that the Mixed Models (that trained on a combined maps of both datasets), have quite similar performance comparing to Australia on Australia and Quest on Quest models. Models on Australia datasets suffers from an over-fitting problem. In the Australia on Australia plot for example, there is a gap between test error and train error when window size is small. Increasing window size helps the model.

Figs. 12 and 13 compare the results of the models on local area (i.e. the area where the model was trained on).

### 5.1. A graphical information system applet

To facilitate the process of interrogating prediction maps, we have created a specialised applet for visualization. Although the models could be evaluated using probability maps and error measurements alone (numerically and statistically), geologists want to compare output probabilities and lines with interpreted lines visually. This relates to the expert domain knowledge of geoscientists, who are trained to recognize textures (e.g., of faults) as geologically plausible or in-plausible. Geoscientists also want to see combinations of input layers underlying our

interpretations. In the GIS visualization applet, the user can:

- load both a model and a dataset
- choose underlying images.
- compare probability map and interpreted lines.
- set a threshold for probabilities
- run the post-processing methods (i.e. clustering and curve fitting).

Fig. 14 shows some screenshots of the applet in different stages with different setups. In this figure, after opening the applet, the user should open the dataset, the probability map and the trained model. Then applet loads all of them (top-left figure). The labeling threshold is a value between 0 and 100 which is divided by 100 to get a probability. The applet uses this threshold to remove all probabilities below the threshold and only shows values above this threshold. By selecting the “show probability map” checkbox, this threshold is ignored, and the predictions will be more or less transparent based on their probability. User can also decide on the color of probabilities in the training and testing areas (bottom-left figure). By default, the underlying layer is a solid black color that user can change that by choosing options from list of input images or by choosing another solid color (top-right figure). Lines color refers to the color of the interpreted-by-expert lines. User can turn on or off this overlay layer. “Run DBSCAN” button calls our modified DBSCAN method, which is the first step in our post-processing routine. User can set  $\epsilon$  and  $MinPts$  values manually in the applet (middle-down figure). User can convert clusters into polynomials by clicking on the “Convert to Lines” button (bottom-right figure). The code for this

applet is publicly available in a GitHub repository.<sup>1</sup>

## 6. Model predictions results in a geologically similar region

**Fig. 15** shows the final output of this model running on a geologically similar region, Swayze area. We collected and generated the same input layers used during the training process for this region, and just run the model to make prediction on this area. That means, the model has never seen any sample from any input layer of this area.

## 7. Discussion and conclusion

In this study we present deep learning, trained using expert's interpretation of lineaments from magnetic and gravity geophysics data, as well as digital elevation data. As would be expected, given machine learning rules-of-thumb, prediction results are more precise and accurate in areas represented by hold-out data in the train/test regions of the model as compared to wholly new geological regions (away from areas used for training). This can be observed in the generally good prediction results in **Figs. 12 and 13**. In contrast, results shown in 15 are generally poor.

**Fig. 15** shows prediction results for the Swayze area. Although the model could label some faults correctly, it generates some false-positives (some faults that are not existing). Results indicate that a generalized lineament model is not created simply, using data and a resultant model from a region with good validation metrics. For all prediction maps (areas of CNN train/test and also the Swayze area) there are clear uncertainties. For example, it is not clear what deficit in the training data results in poor prediction in new areas. One possibility is that the DTM data (representing elevation as a magnitude) is not abstracted within the CNN architecture sufficiently to become generalizable ([Scheiber et al. \(2015\)](#)). If so, feature engineering would assist to create a more robust model (e.g., using a derived layer from the DTM, such as curvature indices). As well, the confidence of input lineament data is uncertain. For example, have interpreted faults for Lock Lilly-Kars and Quest been robustly validated in the field, and are they accurate?

The CNN model, was designed and trained with one target feature (i.e. the existence of a lineament in a location) in [Aghaee Rad \(2019\)](#). This model certainly will be improved in the future by adding secondary features like learning the orientation of a lineament in a location. Having the orientation feature for a lineament location helps the model in making more accurate predictions for nearby locations. Moreover, the dataset itself plays an important role in obtaining a well-trained model. As more lineaments and training data from new regions is integrated, we anticipate and increasingly accurate and generalizable model.

## CRediT authorship contribution statement

**Amin Aghaee:** Conceptualization, of this study and, Software, software development; Programming and debugging, Data curation, Data, Formal analysis, Analysis and Data Interpretation, Writing – original draft, Writing. **Pejman Shamsipour:** Programming and debugging, Data, Formal analysis, Analysis and Data Interpretation and, Data curation, Writing – review & editing, Editing. **Shawn Hood:** Geological Data Interpretation and, Data curation, Writing and, Writing – review & editing, Editing; Figure Preparation. **Rasmus Haugaard:** Writing – original draft, Writing and, Writing – review & editing, editing; Research Associate at Metal Earth and Swayze area, Project administration, project leader.

## Declaration of competing interest

The authors declare that they have no known competing financial

interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgement

The first author would like to thank Professor David Poole for introducing this research topic and for his invaluable consultation on various aspects of this work. As well, to thank the Mitacs organisation and Minerva Intelligence Ltd. for sponsorship and data preparation assistance during Masters studies. The authors wish to express gratitude to Thomas Bissig (Bissig Geoscience) for his insight related to confidence on the interpretation of regional faults in British Columbia. We also appreciate information from Mark Rattenbury (Geologist and Programme Leader at GNS Science in New Zealand) regarding the public geophysics data in New Zealand. Finally, we thank Laurentian University and Metal Earth Project (MERC-ME-2021-020).

## References

- Aghaee Rad, M.A., 2019. Machine Learning of Lineaments from Magnetic, Gravity and Elevation Maps. MSc. thesis, University of British Columbia.
- Ayer, J., Amelin, Y., Corfu, F., Kamo, S., Ketchum, J., Kwok, K., Trowell, N., 2002a. Evolution of the southern abitibi greenstone belt based on u-pb geochronology: autochthonous volcanic construction followed by plutonism, regional deformation and sedimentation. *Precambrian Res.* 115, 63–95.
- Ayer, J., Trowell, N., Valade, L., Nevills, M., Madon, Z., 2002b. Geological Compilation of the Swayze Area. Abitibi Greenstone Belt, p. 3511.
- Baatar, B., Parra-Avila, L., Fiorentini, M., Polito, P., Crawford, A., 2020. Porphyry copper fertility of the loch lilly-kars belt, western new south wales, Australia. *Aust. J. Earth Sci.* 67, 75–87.
- Brathwaite, R., 1974. The geology and origin of the rosebery ore deposit, tasmania. *Econ. Geol.* 69, 1086–1101.
- Chandra, S., Rao, V., Krishnamurthy, N., Dutta, S., Ahmed, S., 2006. Integrated studies for characterization of lineaments used to locate groundwater potential zones in a hard rock region of Karnataka, India. *Hydrogeol. J.* 14, 1042–1051.
- Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images. In: *Advances in Neural Information Processing Systems*, pp. 2843–2851.
- Cooper, G., Cowan, D., 2006. Enhancing potential field data using filters based on the local phase. *Comput. Geosci.* 32, 1585–1591.
- Corbett, K., 1992. Stratigraphic-volcanic setting of massive sulfide deposits in the cambrian mount read volcanics, tasmania. *Econ. Geol.* 87, 564–586.
- Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd*, pp. 226–231.
- Geiß, C., Taubenböck, H., 2013. Remote sensing contributing to assess earthquake risk: from a literature review towards a roadmap. *Nat. Hazards* 68, 7–48.
- Green, G., Solomon, M., Walshe, J., 1981. The formation of the volcanic-hosted massive sulfide ore deposit at rosebery, tasmania. *Econ. Geol.* 76, 304–338.
- Greenfield, J.E., Lees, T.C., Parr, J.M., McConachy, T.F., 2003. A review of broken hill ore system models. *Geosci. Aust. Rec.* 13, 57–60.
- Heather, K., Shore, G., van Breeman, O., 1995. The convoluted layer-cake: an old recipe with new ingredients for the swayze greenstone belt, southern superior province, Ontario. *Curr. Res. 1–10.*
- Hobbs, W.H., 1904. Lineaments of the atlantic border region. *Bull. Geol. Soc. Am.* 15, 483–506.
- Isles, D.J., Rankin, L.R., 2013. Geological Interpretation of Aeromagnetic Data. Society of Exploration Geophysicists and Australian Society of Exploration.
- Katz, L.R., Kontak, D.J., Dubé, B., McNicoll, V., 2017. The geology, petrology, and geochronology of the archean coté gold large-tonnage, low-grade intrusion-related au (-cu) deposit, swayze greenstone belt, ontario, Canada. *Can. J. Earth Sci.* 54, 173–202.
- Kocal, A., Duzgun, H.S., Karpuz, C., 2004. Discontinuity Mapping with Automatic Lineament Extraction from High Resolution Satellite Imagery. *ISPRS XX*, Istanbul, pp. 12–23.
- Krause, J., Jin, H., Yang, J., Fei-Fei, L., 2015. Fine-grained recognition without part annotations. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5546–5555.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105.
- Lary, D.J., Alavi, A.H., Gandomi, A.H., Walker, A.L., 2016. Machine learning in geosciences and remote sensing. *Geosci. Front.* 7, 3–10.
- Lillesand, T., Kiefer, R.W., Chipman, J., 2015. *Remote Sensing and Image Interpretation*. John Wiley & Sons.
- Liu, Y., Li, Z.X., Laukamp, C., West, G., Gardoll, S., 2013. Quantified spatial relationships between gold mineralisation and key ore genesis controlling factors, and predictive mineralisation mapping, st ives goldfield, western Australia. *Ore Geol. Rev.* 54, 157–166.

<sup>1</sup> <https://github.com/aminrd/LineamentLearning>.

- Ltd, G., 2007a. Report on a Helicopter-Borne Versatile Time Domain Electromagnetic (VTEM) and Horizontal Magnetic Gradiometer Geophysical Survey: Quest Project Central British Columbia Nts 93a,b,g,h,j,k,n,o & 94c,d. Project 7042.
- Ltd, S.G., 2007b. Airborne Gravity Survey Quesnellia Region british columbia 2008.
- Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., Jie, W., 2015. Remote sensing big data computing: challenges and opportunities. Future Generat. Comput. Syst. 51, 47–60.
- Masoud, A., Koike, K., 2017. Applicability of computer-aided comprehensive tool (linda: lineament detection and analysis) and shaded digital elevation model for characterizing and interpreting morphotectonic features from lineaments. Comput. Geosci. 106, 89–100.
- Middleton, M., Schnur, T., Sorjonen-Ward, P., Hyvönen, E., 2015. Geological lineament interpretation using the object-based image analysis approach: results of semi-automated analyses versus visual interpretation. Geol. Surv. Finland, Special Paper 57, 135–154.
- Paine, D.P., Kiser, J.D., 2012. Aerial Photography and Image Interpretation. John Wiley & Sons.
- Pilkington, M., Keating, P., 2004. Contact mapping from gridded magnetic data? a comparison of techniques. Explor. Geophys. 35, 306–311.
- Ramli, M., Yusof, N., Yusoff, M., Juahir, H., Shafri, H., 2010. Lineament mapping and its application in landslide hazard assessment: a review. Bull. Eng. Geol. Environ. 69, 215–233.
- Ran, L., Zhang, Y., Wei, W., Zhang, Q., 2017. A hyperspectral image classification framework with spatial pixel pair features. Sensors 17, 2421.
- Ray, R.G., 1960. Aerial Photographs in Geologic Interpretation and Mapping, vol. 373. US Govt. Print. Off.
- Reichheld, S., 2012. Documentation and assessment of exploration activities generated by geoscience bc data publications. In: Geoscience BC Summary of Activities, 2013–1.
- Rogers, J., Beilhartz, D., Kontak, D., Katz, L., Dubé, B., McNicoll, V., 2013. The côté gold deposit: discovery of a new generation low-grade, multi-million ounce gold resource in the archean superior province of Canada. In: NewGenGold 2013 Conference; Perth; AU; November 27, 2013, NewGenGold 2013 Conference.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.
- Sander, P., 2007. Lineaments in groundwater exploration: a review of applications and limitations. Hydrogeol. J. 15, 71–74.
- Scheiber, T., Fredin, O., Viola, G., Jarna, A., Gasser, D., Łapińska-Viola, R., 2015. Manual extraction of bedrock lineaments from high-resolution lidar data: methodological bias and human perception. GFF 137, 362–372.
- Sharp, T.R., R.D.R.H.M.S.M.K.J.S.B.P.J., 2006. Loch Lilly-Kars Belt 1:250 000 Geophysical-Geological Interpretation Map, first ed.
- Spanhol, F.A., Oliveira, L.S., Petitjean, C., Heutte, L., 2016. Breast cancer histopathological image classification using convolutional neural networks. In: 2016 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 2560–2567.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.
- Tiren, S., 2010. Lineament Interpretation. Short Review and Methodology. Technical Report. Swedish Radiation Safety Authority.
- Tudor Ionescu, R., Alexe, B., Leordeanu, M., Popescu, M., Papadopoulos, D.P., Ferrari, V., 2016. How hard can it be? estimating the difficulty of visual search in an image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2157–2166.
- Vassilas, N., Perantonis, S., Charou, E., Tsengoglou, T., Stefoiu, M., Varoufakis, S., 2002. Delineation of lineaments from satellite data based on efficient neural network and pattern recognition techniques. In: Proc. Second Hellenic Conference on AI. Citeseer, pp. 355–365.
- Woodward, O.H., 1965. A Review of the Broken Hill Lead-silver-zinc Industry, Produced by the Broken Hill Mining Managers' Association,(and) the Australasian Institute of Mining and Metallurgy. West Publishing Corporation.
- Xie, L., Wang, J., Wei, Z., Wang, M., Tian, Q., 2016. Disturlabel: regularizing cnn on the loss layer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4753–4762.
- Xu, Q., Zhang, M., Gu, Z., Pan, G., 2019. Overfitting remedy by sparsifying regularization on fully-connected layers of cnns. Neurocomputing 328, 69–74.