Journal of Geophysical Research

Supporting Information for

Automatic fault mapping in remote optical images and topographic data with deep learning

Lionel MATTÉO[1], Isabelle MANIGHETTI[1], Yuliya TARABALKA[2], Jean-Michel GAUCEL[3], Martijn VAN DEN ENDE[1], Antoine MERCIER[1], Onur TASAR[4], Nicolas GIRARD[4], Frédérique LECLERC[1], Tiziano GIAMPETRO[1], Stéphane DOMINGUEZ[5], and Jacques MALAVIEILLE[5]

1 Université Côte d'Azur, Observatoire de la Côte d'Azur, IRD, CNRS, Géoazur, Sophia Antipolis, France

2 LuxCarta, Sophia Antipolis, France

3 Thales Alenia Space, Cannes, France

4 Université Côte d'Azur, Inria, Sophia Antipolis, France

5 Université de Montpellier, Géosciences Montpellier, France

Contents of this file

Additional Supporting Information

Text S1 (with Fig. A1): detailed description of the deep learning approach followed in present study.

Fig. S2: (a) Cross-entropy loss function evolution with number of epoch calculations. One epoch corresponds to 1000 iterations where 12 random images from the training dataset pass inside the model at each iteration. While the validation loss might be thought to increase from about 19 epochs, it starts increasing in a more steady fashion from about 44 epochs. (b) the M_{Ref} prediction at 19 epochs (sub-figure d) is less accurate than that at 44 epochs (sub-figure c). We have thus stopped the training at 44 epochs.

Fig. S3: Tversky index for the models M_{A1} to M_{A7}. The two horizontal lines in (a) are the TI values for M_{Ref} with respect to basic or refined mapping.

Fig. S4: Comparison of predictions in site A calculated with M_{A3} (b) (two filters only in first layer), and with M_{Ref} (d) (64 filters in first layer). The M_{Ref} predictions better compare to the refined mapping (c) and the image (a). Red square in (a) locates Figs. 5, 6, 8.

Fig. S5: Comparison of predictions in site A calculated with M_{A7} (b) (32 filters in first layer), and with M_{Ref} (d) (64 filters in first layer). Figure S5A shows entire validation zone, while Fig. S5B shows a zoom highlighting the small features. The M_{Ref} predictions better compare to the refined mapping (c) and the image (a). Red square in Fig. S5Aa locates Figs. S5B, 5, 6, 8.

Fig. S6: Tversky index for the models M_{T1} to M_{T7}. M_{T6} is equivalent to M_{Ref}. The two horizontal dotted lines are the TI values obtained with Canny edge filter and GVG detector algorithms.

Fig. S7: Impact of training data size: Entire validation zone of site A is shown. (a) and (b) as in Fig. 5. (d) shows predictions from M_{Ref} (i.e., 100% of the available training data), while (e)-(i) show predictions with decreasing amount of training data (from 75 to 5%), and (c) predictions with additional training data (from site C). See text for details. Red square in (a) locates Figs. 5, 6, 8.

Fig. S8: Impact of training data size: zooms from validation zone of site A. Same as in Fig. S7.

Fig. S9: Comparison of predictions in site A when M_{Ref} is trained with refined mapping (shown in c) which produces model M_{Q1} (shown in b), or with basic mapping (shown in d) which produces model M_{Q2} (shown in e). The predictions of M_{Q1} are richer than those of M_{Q2} and recover fine tectonic features, some are indicated (R: relay zone, E: en echelon segments, I: inner damage). Red square in (a) locates Figs. 5, 6, 8.

Fig. S10: ROC curves (Receiving Operating Characteristic) for all calculated models. The ROC metric compares the "False positive rate" (predicted fault/fracture location not present in the ground truth) to the "True positive rate" (predicted fault/fracture location present in the ground truth) at increasing probability thresholds. With this metric good model performance is characterized by a low False positive rate and a high True positive rate.

Fig. S11: M_{Ref} predictions (c) in site B (entire site, ground photogrammetry), compared to basic mapping ground truth (b), and to Canny edge (e) and GVG detector (f) results. Thinned predictions (probability > 0.9) are also shown (d). Red square in (a) indicates Fig. 12.

Fig. S12: Model predictions in entire site E (drone photogrammetry shown in a), with comparison between M_{Ref} trained only on sites A and B (b), and M_{Ref} enriched with a few transfer learning from site D with drone photogrammetry (c). The transfer learning improves the predictions. A zoom of the figure is shown in Fig. 15.

Fig. S13: Model predictions in entire site G (Pléiades satellite image shown in a), with comparison between M_{Ref} trained only on sites A and B (b), and M_{Ref} enriched with a few transfer learning from site F with Pléiades data (c). The transfer learning improves the predictions.

Fig. S14: Model predictions in entire site C (ground photogrammetry shown in a), when M_{Ref} is trained with topography (c) and without topography (d). Red square in (a) indicates Fig. 14 (training with topography).

Fig. S15: Model predictions in entire site E (drone photogrammetry shown in a), when M_{Ref} is trained with topography (b) and without topography (c).

Mattéo et al. - Supplementary Text S1

Deep Learning and its associated Artificial Neural Networks (ANNs) represent a class of Machine Learning algorithms that provide an approximate mapping f from an input y to an output $\hat{y}$, parametrised by a sets of "weights" and "biases" (jointly denoted by θ), i.e. $\hat{y}$: $\hat{y} = f_\theta(y)$. In a supervised setting, the goal is to optimise some arbitrarily defined cost function $\mathcal{L}$ of the model output $\hat{y}$ and a ground truth y, for the parameters θ, i.e.:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}(\hat{y}_i, f_\theta(y_i))$$

Where the summation is performed over N input-ground truth pairs $(y_i, \hat{y}_i)$ in the data set. An ANN is represented as a stack of layers $l_j, j \in [1..m]$, such that $f_\theta = l_m \circ l_{m-1} \circ \ldots \circ l_1$. In turn, each layer is usually a simple parametrised operation, commonly taken as a matrix multiplication and addition: $l(x) = Wx + b$, with W being a matrix of "weights", and b the "bias" vector (though in principle W and b can be of arbitrary rank). This multiplication is followed by a non-linear function called the "activation function" to enable f_θ to represent non-linear mappings. Various activation functions have been proposed, but most often a piece-wise linear function, the Rectified Linear Unit (ReLU), is adopted: $\mathrm{ReLU}(y) = \max(0, y)$. Since all operations in the ANN are differentiable, θ^* can be obtained through gradient-based optimisation of Eq. (1).

One of the most basic configurations, the Multi-Layer Perceptron (Rosenblatt, 1957), features a dense weight matrix W that relates every entry of the input vector y to every entry in the layer output vector l. For this reason, Perceptron layers are sometimes referred to as "fully-connected" or "dense" layers. However, in many data types, such as time series and images, the data are only locally correlated, rendering distant data points (e.g. pixels that are situated at the far ends of the image) essentially independent of each other. Convolutional Neural Networks (CNNs; Fukushima, 1980; LeCun et al., 2015) leverage this property by only performing the multiplications within a small window called the "kernel". The same kernel weights are re-used for every location in the data set, which can be envisioned as convolving the input data with a kernel of fixed size. This convolution (or more precisely: cross-correlation) operation is represented by a matrix multiplication, effectively rendering W into a structured and sparse matrix (see Fig.A1 below). Moreover, W may comprise multiple kernels (or "features"), such that each kernel produces a "feature map" that is proportional in spatial extent to the input. After passing through an activation function, subsequent convolutional layers takes these feature maps as an input, so that the original input (e.g. a 3-channel R/G/B image) is transformed into a complex output with an arbitrary number of channels. Typical kernel sizes that

are used in image analysis are 3x3 or 5x5 pixels (see Fig. A1), which implies that each individual kernel only has a narrow perceptive field (i.e. it sees only a small part of the input data). Fortunately, by stacking numerous convolutional layers, the perceptive field gradually expands throughout the CNN, so that deep CNNs exhibit a sufficiently wide perceptive field for e.g. object localisation and identification.

To further enhance the model's perceptive field, and to reduce the number of computational operations, the input data is often gradually down-sampled through "pooling" operations (see Fig. A1 below). The output of a convolutional layer is similar in spatial extent as the layer input, which can be reduced by e.g. taking the maximum value of each group of 2x2 pixels. The resulting output of the pooling operation is then only half in extent in each dimension, and this down-sampled output serves as the input to the next convolutional layer.

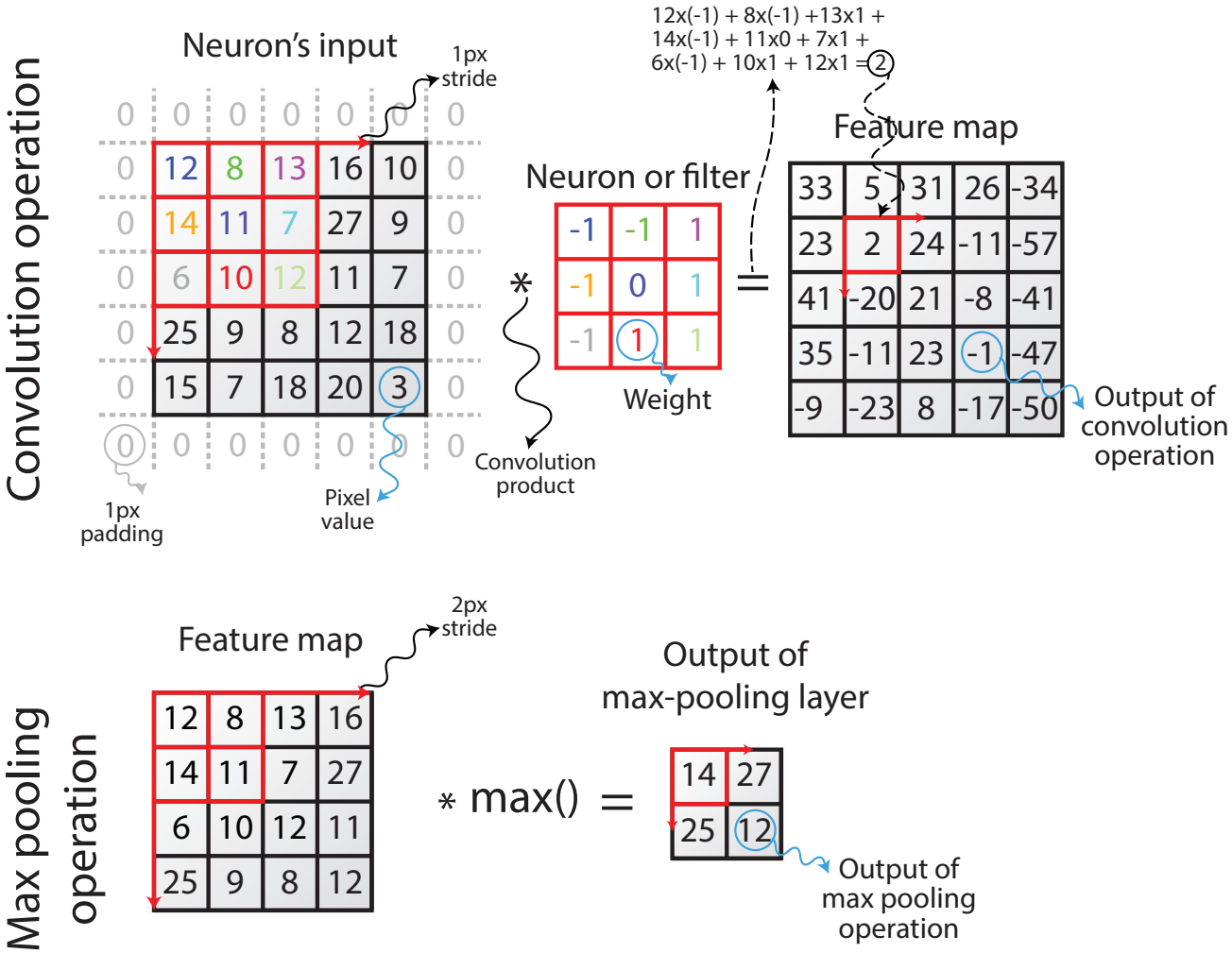


Figure A1

Fig. S2a

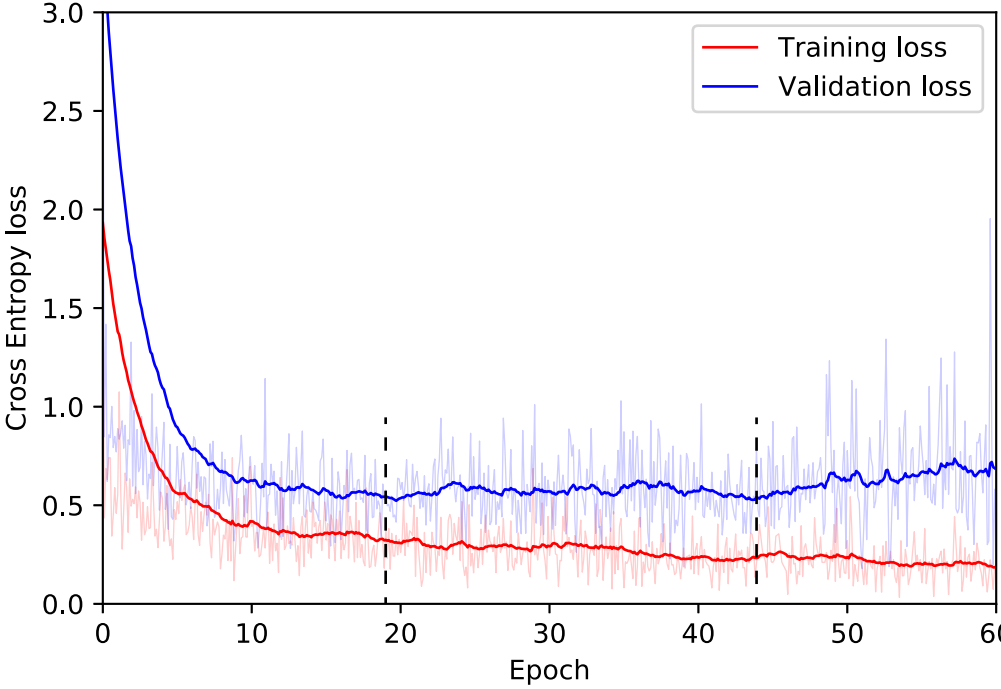

Training loss
Validation loss
Cross Entropy loss
Epoch
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
10
20
30
40
50
60

Fig. S2b

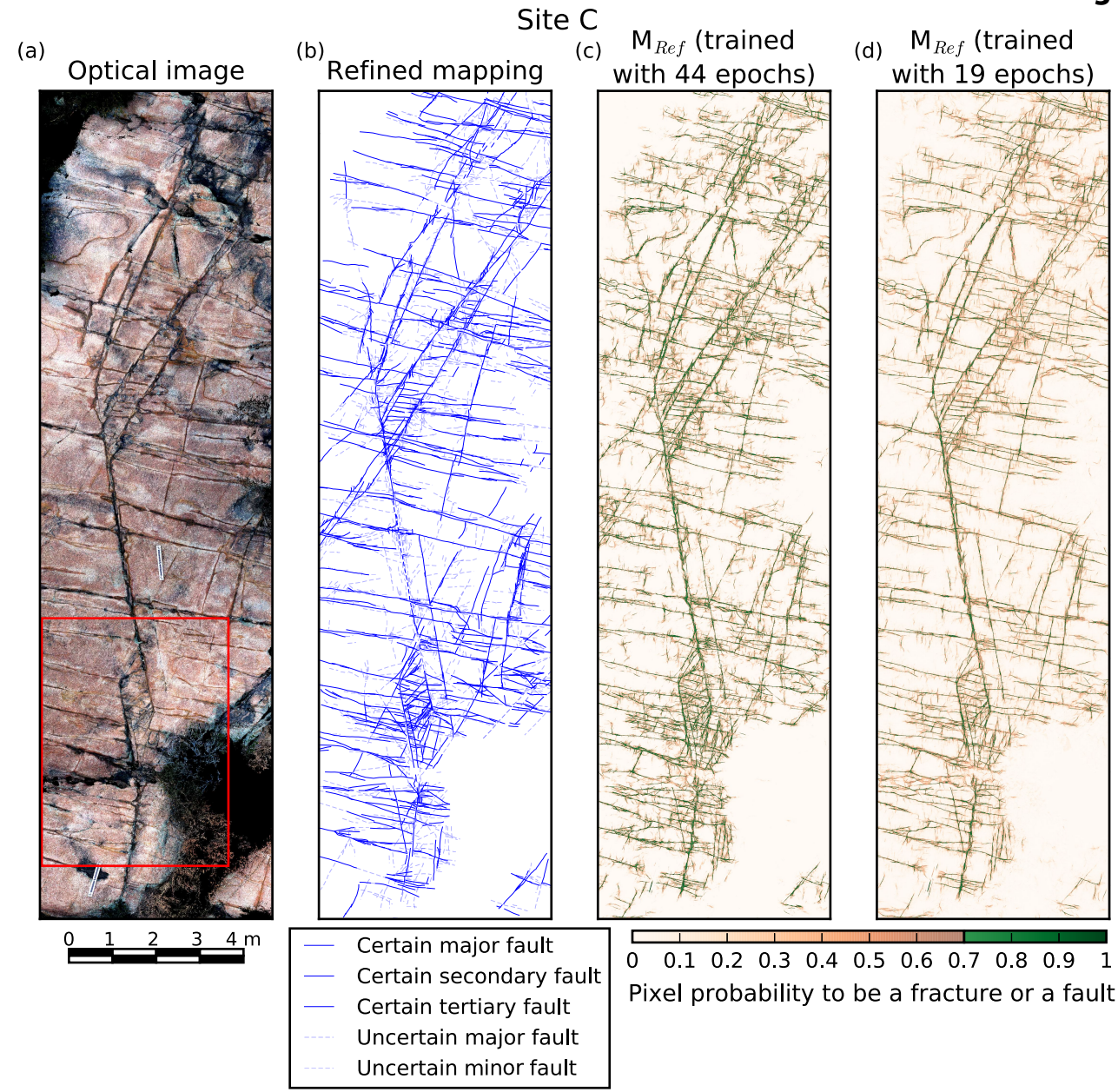

Site C
(a)
Optical image
(b)
Refined mapping
(c)
M_{Ref} (trained with 44 epochs)
(d)
M_{Ref} (trained with 19 epochs)
0 1 2 3 4 m
Certain major fault
Certain secondary fault
Certain tertiary fault
Uncertain major fault
Uncertain minor fault
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Pixel probability to be a fracture or a fault

Fig. S3

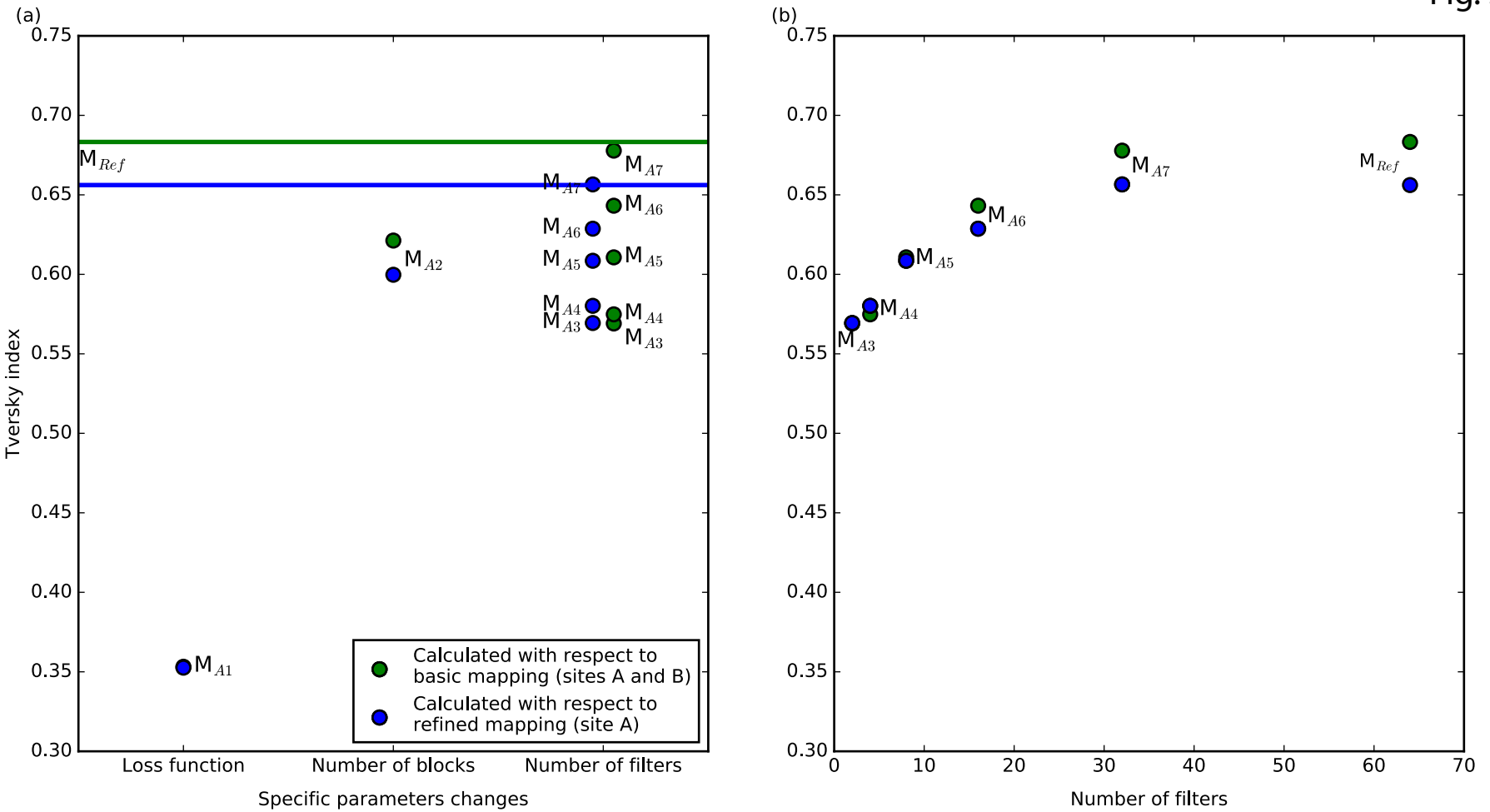

(a)
(b)
Tversky index
Specific parameters changes
Loss function
Number of blocks
Number of filters
Number of filters
Calculated with respect to basic mapping (sites A and B)
Calculated with respect to refined mapping (site A)
M Ref
M A1
M A2
M A3
M A4
M A5
M A6
M A7

Fig. S4

Site A

(a) Optical image

(b) M_{A3} (trained with 2 filters in first layer)

(c) Refined mapping

(d) M_{Ref} (trained with 64 filters in first layer)

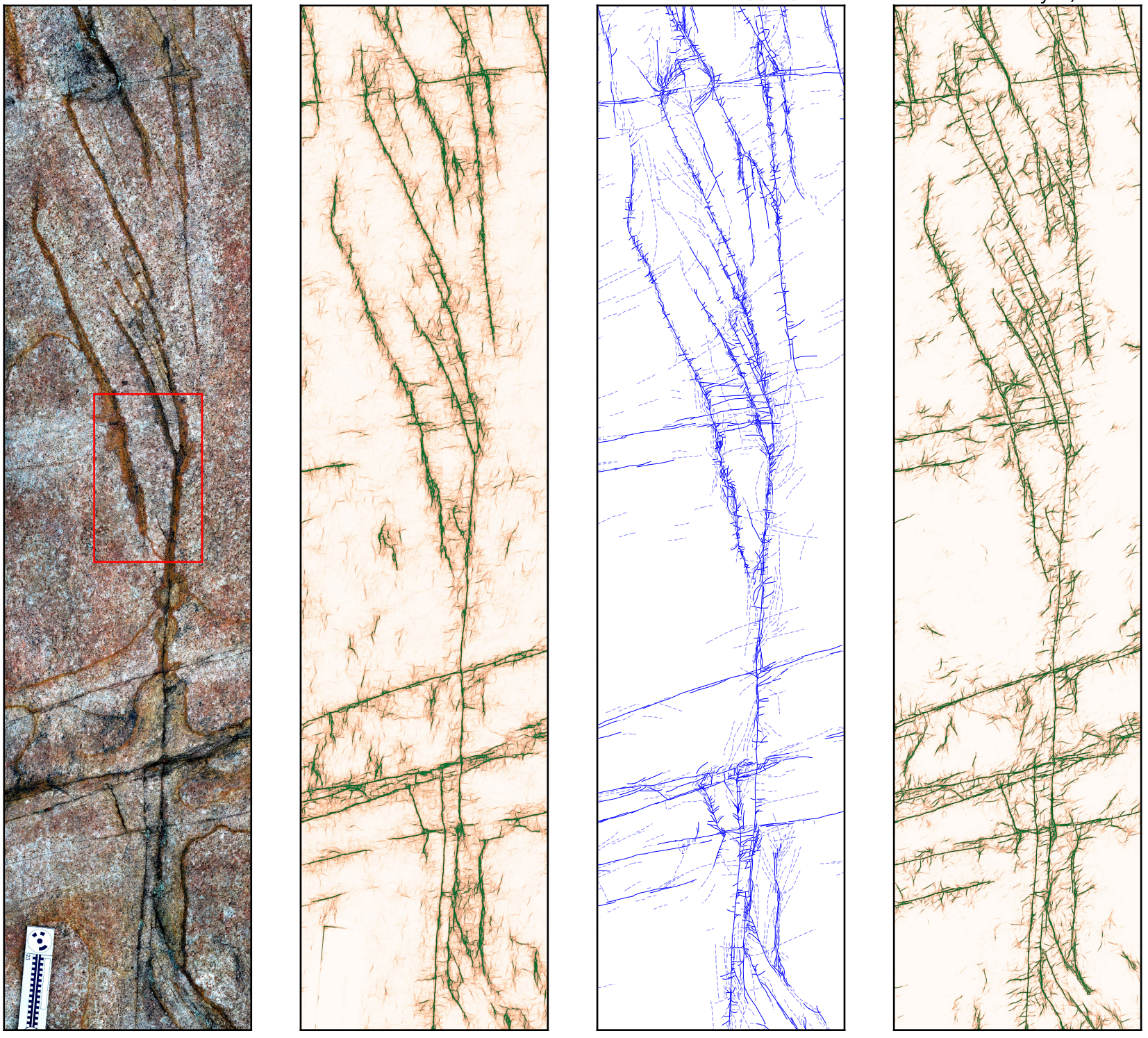


Fig. S5A

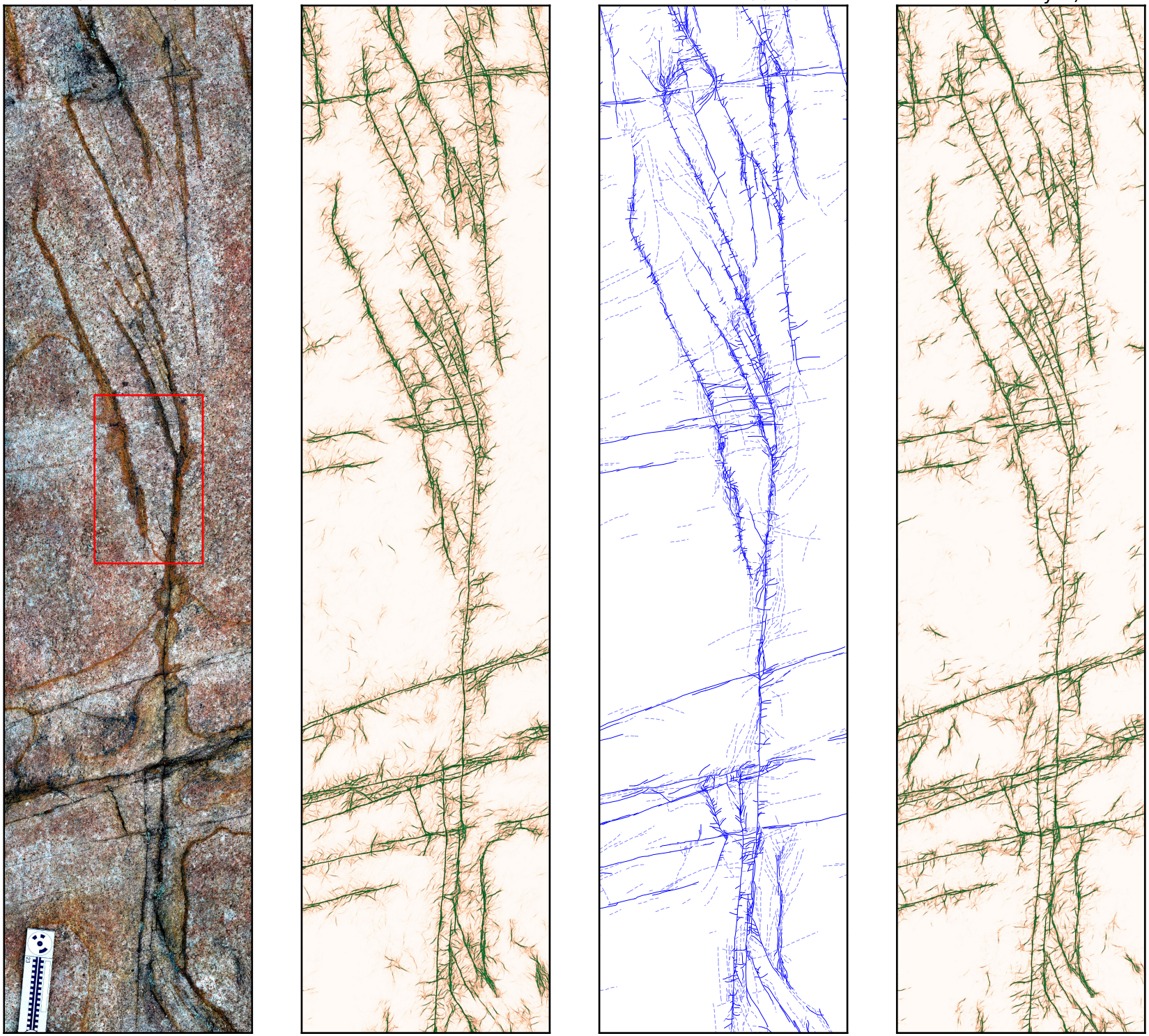
Site A

(a) Optical image

(b) M_{A7} (trained with 32 filters in first layer)

(c) Refined mapping

(d) M_{Ref} (trained with 64 filters in first layer)



Site A

Fig. S5B

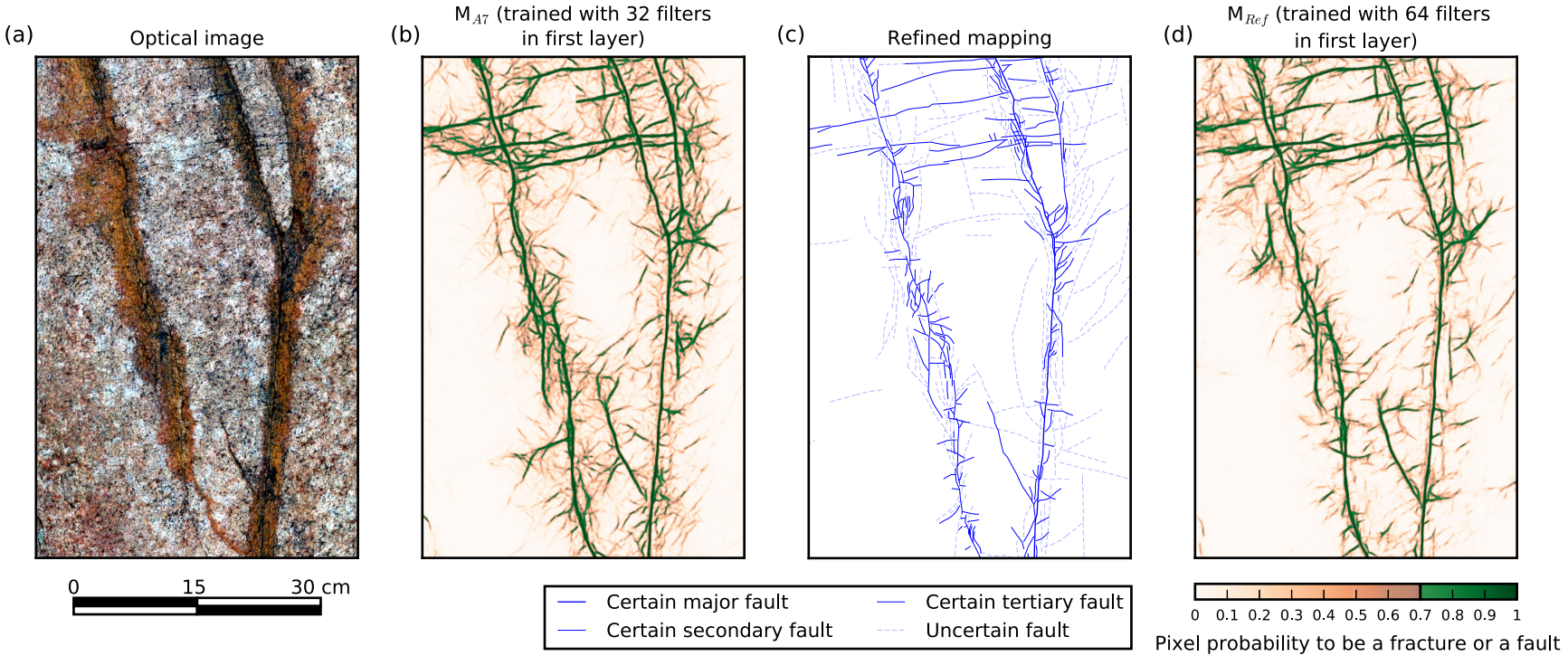
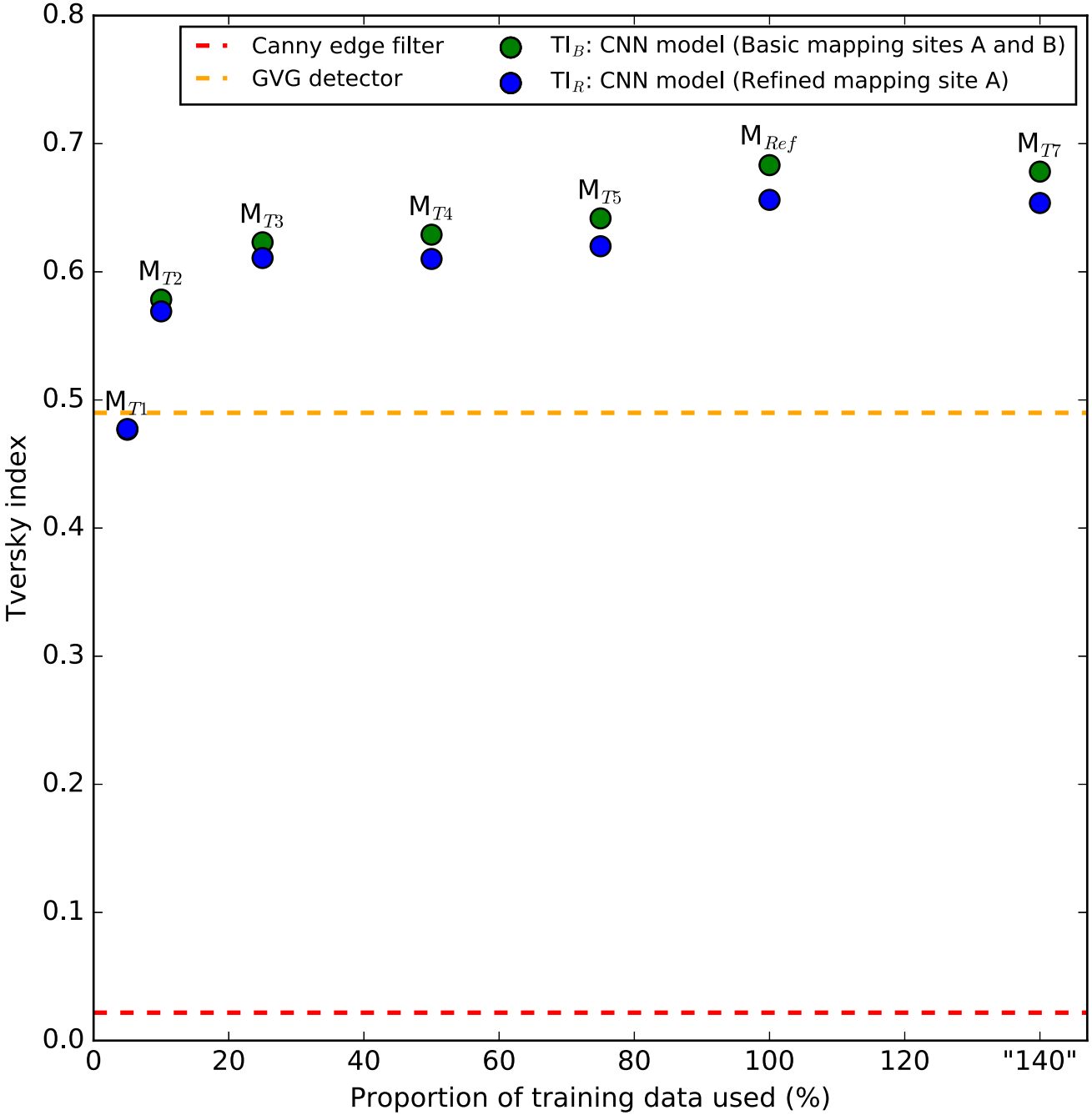
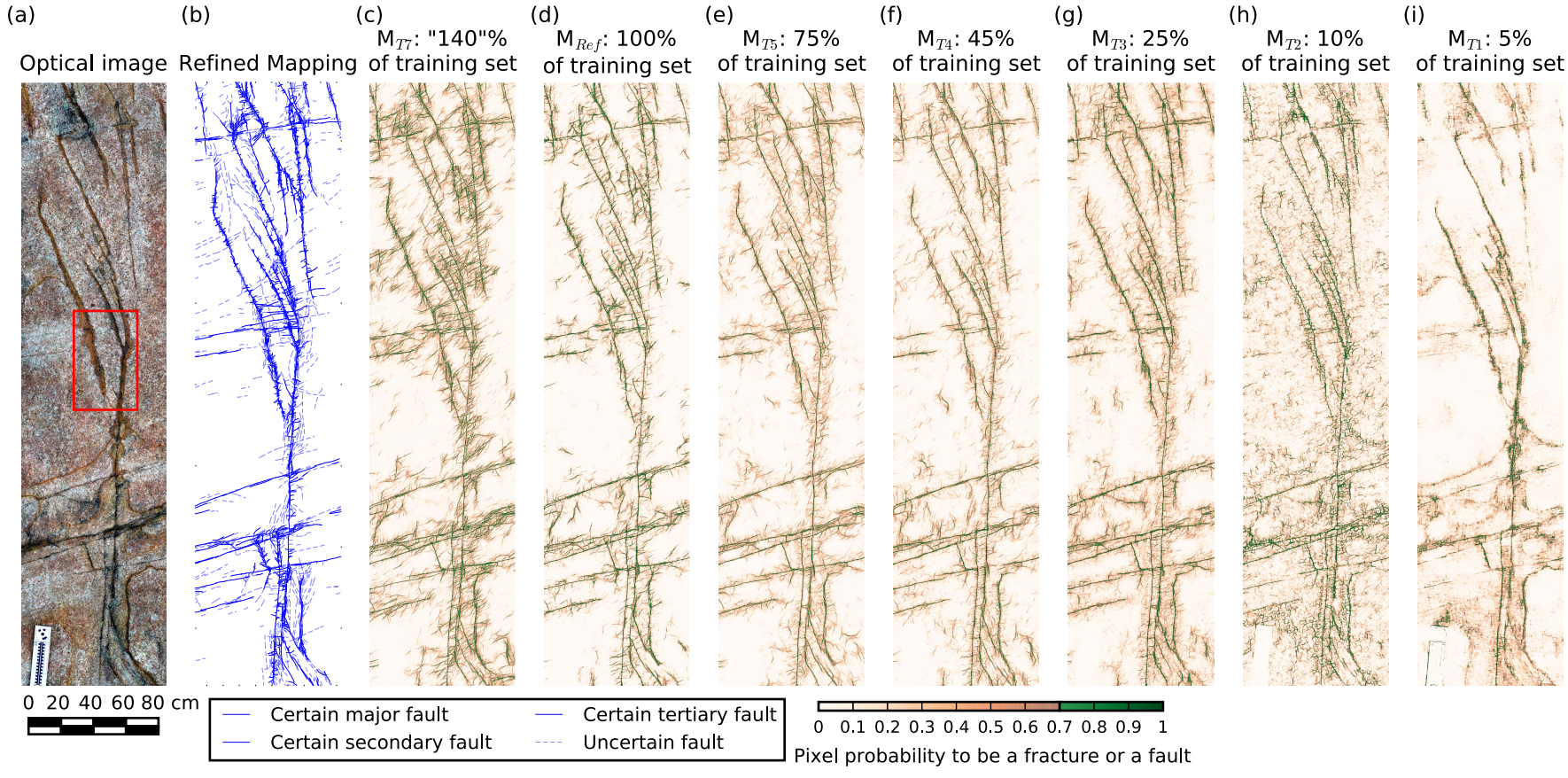
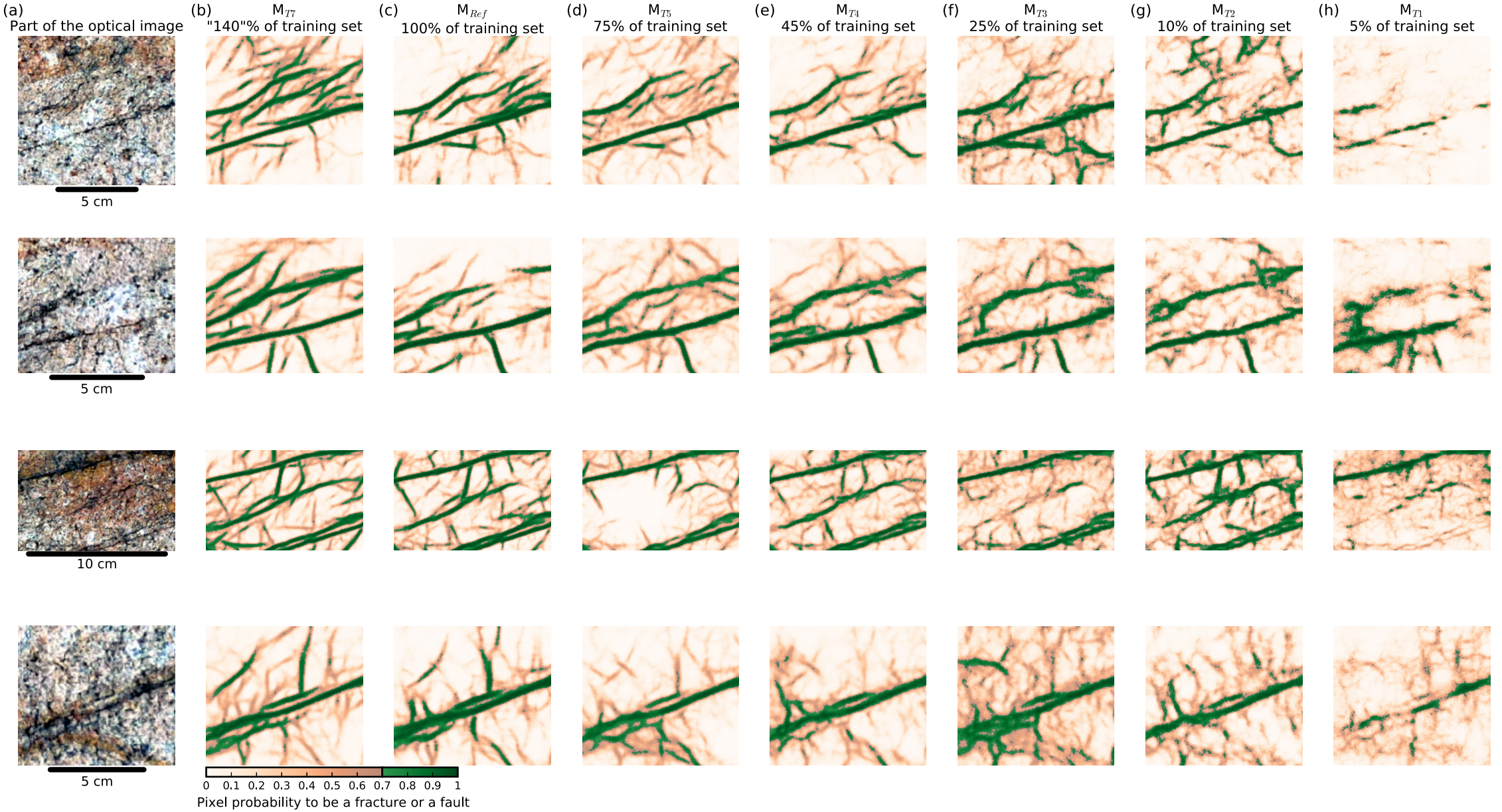


Fig. S6


Canny edge filter
GVG detector
TI$_B$: CNN model (Basic mapping sites A and B)
TI$_R$: CNN model (Refined mapping site A)
M$_{T1}$
M$_{T2}$
M$_{T3}$
M$_{T4}$
M$_{T5}$
M$_{Ref}$
M$_{T7}$
Tversky index
Proportion of training data used (%)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
20
40
60
80
100
120
"140"


Site A
(a)
Optical image
(b)
Refined Mapping
(c)
M_{T7}: "140"% of training set
(d)
M_{Ref}: 100% of training set
(e)
M_{T5}: 75% of training set
(f)
M_{T4}: 45% of training set
(g)
M_{T3}: 25% of training set
(h)
M_{T2}: 10% of training set
(i)
M_{T1}: 5% of training set
0 20 40 60 80 cm
Certain major fault
Certain secondary fault
Certain tertiary fault
Uncertain fault
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Pixel probability to be a fracture or a fault

Site A


(a)
Part of the optical image
(b) M_{T7}
"140"% of training set
(c) M_{Ref}
100% of training set
(d) M_{T5}
75% of training set
(e) M_{T4}
45% of training set
(f) M_{T3}
25% of training set
(g) M_{T2}
10% of training set
(h) M_{T1}
5% of training set
5 cm
5 cm
10 cm
5 cm
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Pixel probability to be a fracture or a fault

Site A

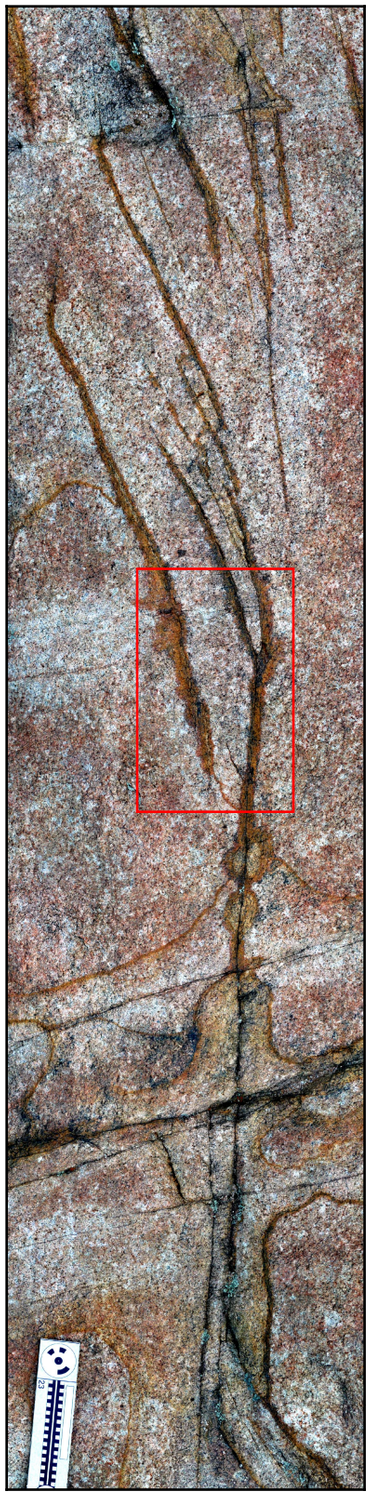
(a) Optical image

(b) MQ1 (training with refined mapping)

(c) Refined mapping

(d) Basic mapping

(e) MQ2 (training with basic mapping)



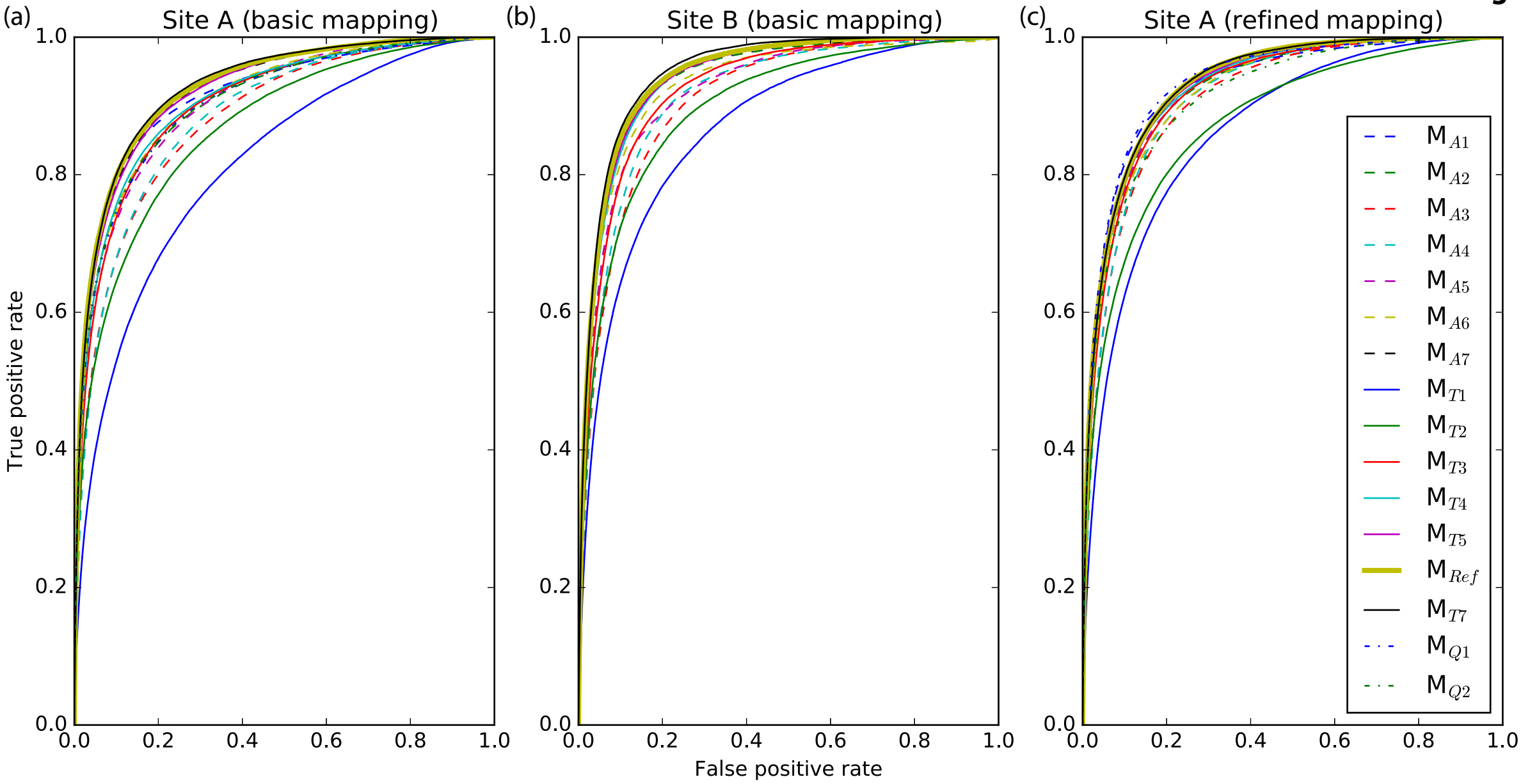
0 20 40 60 80 cm

Certain major fault
Certain secondary fault
Certain tertiary fault
Uncertain fault

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

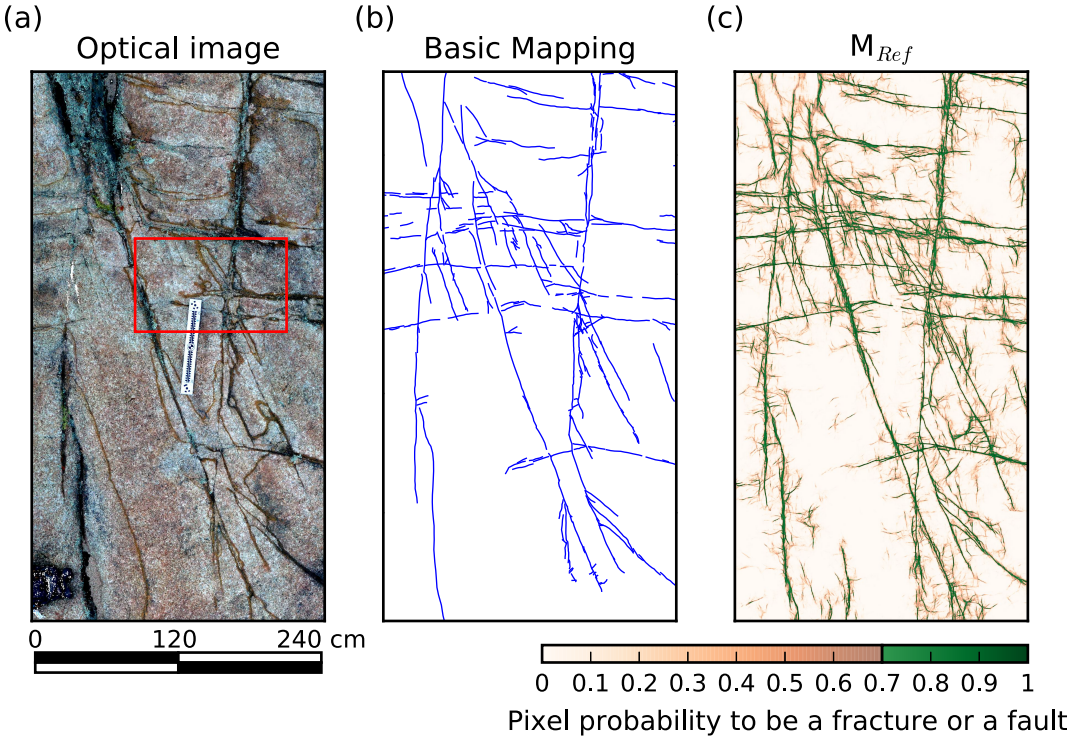
Pixel probability to be a fracture or a fault

Fig. S10


(a)
Site A (basic mapping)
(b)
Site B (basic mapping)
(c)
Site A (refined mapping)
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
M_{A1}
M_{A2}
M_{A3}
M_{A4}
M_{A5}
M_{A6}
M_{A7}
M_{T1}
M_{T2}
M_{T3}
M_{T4}
M_{T5}
M_{Ref}
M_{T7}
M_{Q1}
M_{Q2}

Site B

Fig. S11

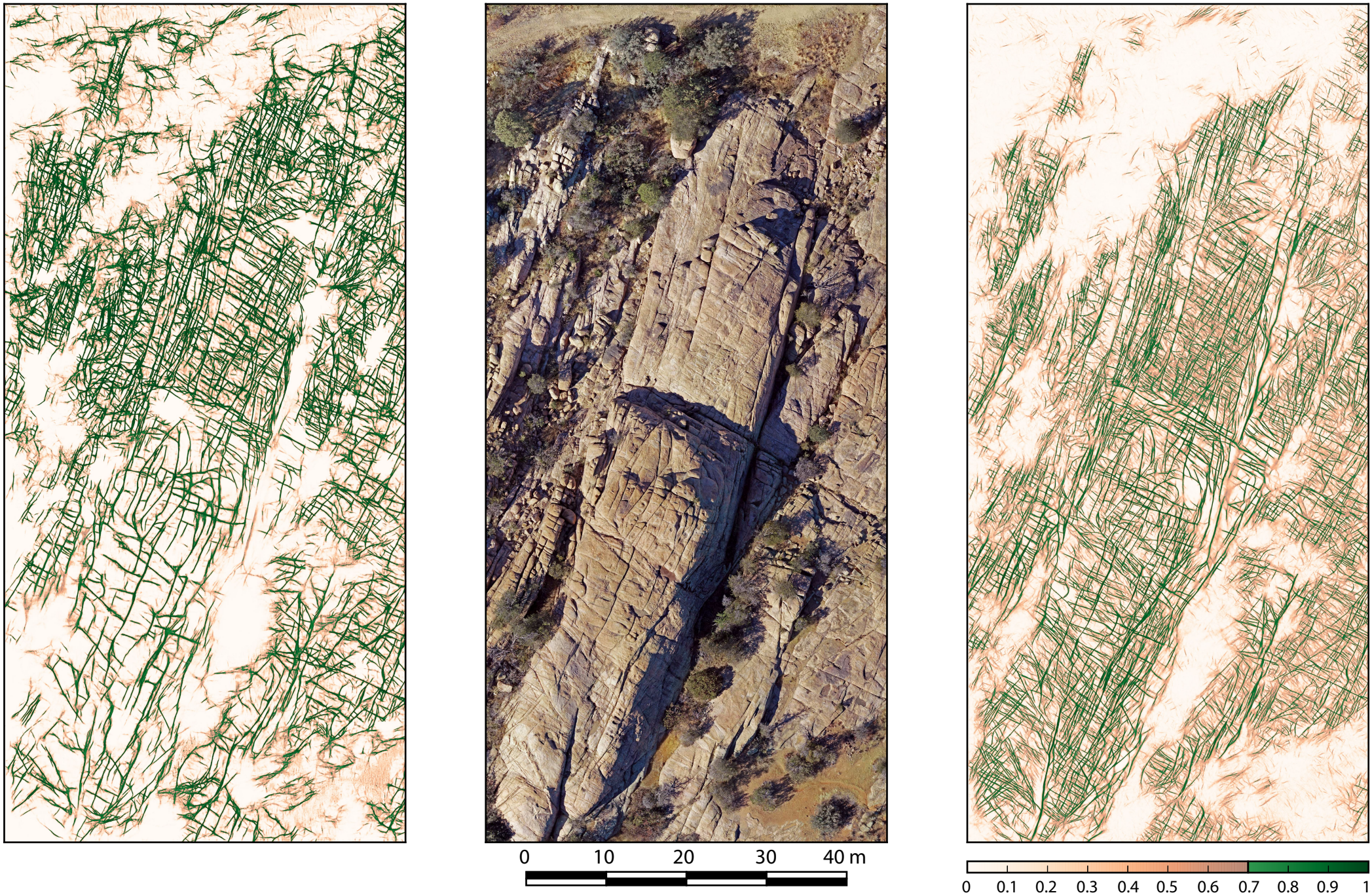


Site E

(b) M_{Ref} (Training on sites A and B)

(a) Optical Drone image

(c) M_{TL1} (Transfer learning from site D)



Site G

(b) M_{Ref} (Training on sites A and B)

(a) Optical Pléiades image

(c) M_{TL2} (Transfer learning from site F)

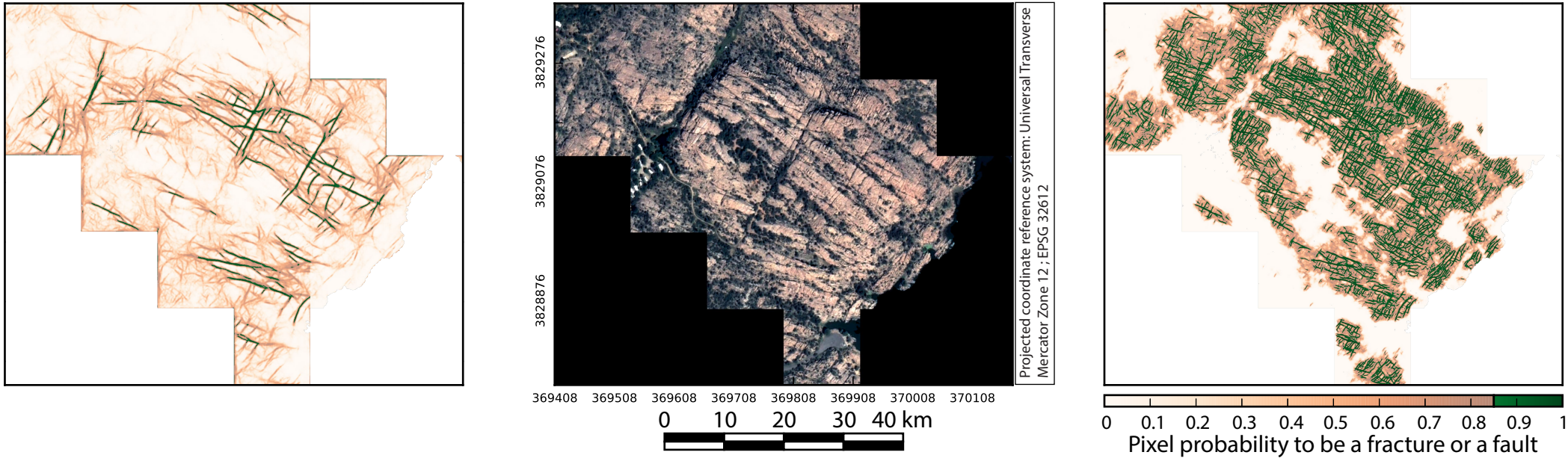


Fig. S14

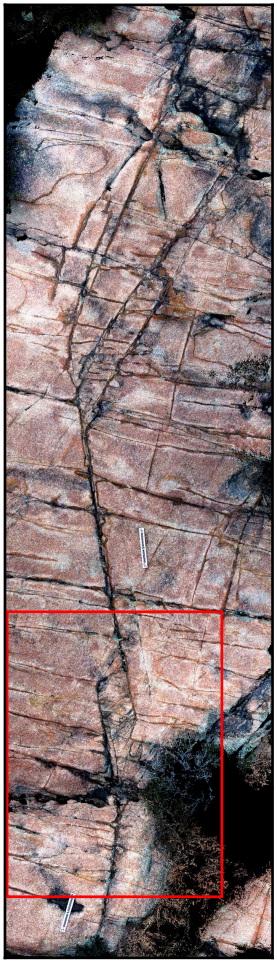
Site C

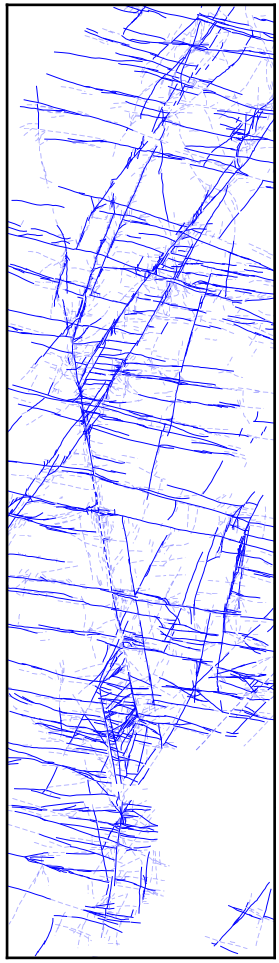
(a) Optical image

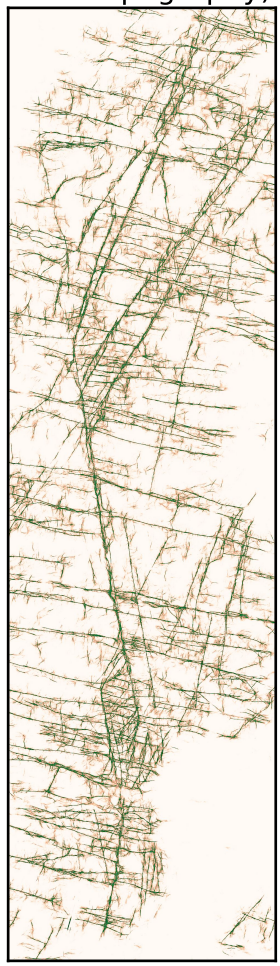
(b) Refined mapping

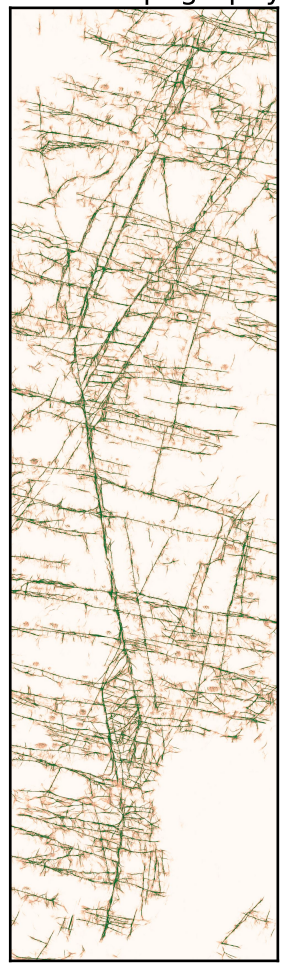
(c) M_{Ref} (trained with topography)

(d) M_{Ref} (trained without topography)









0 1 2 3 4 m

Certain major fault
Certain secondary fault
Certain tertiary fault
Uncertain major fault
Uncertain minor fault

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Pixel probability to be a fracture or a fault

Site E

(b) M_{Ref} (Training on sites A and B with topography)

(a) Optical Drone image

(c) M'_{Ref} (Training on sites A and B without topography)