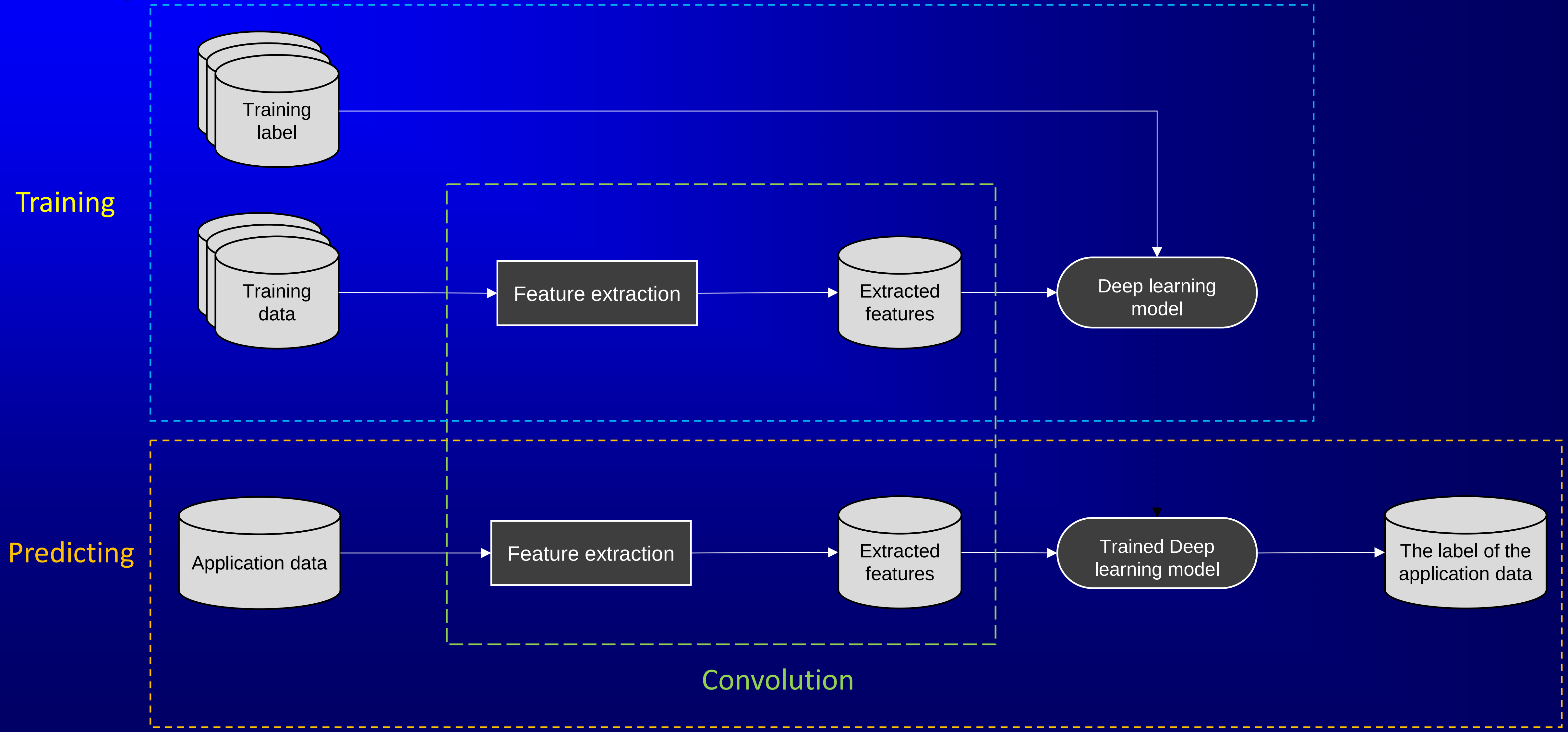
Seismic Attributes - from Interactive Interpretation to Machine Learning

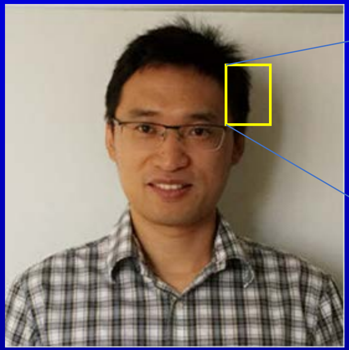
Jie Qi and Kurt J. Marfurt
(The University of Oklahoma)

Deep Learning and CNN fault detection

A deep learning workflow



Convolutional Neural networks



1	0	1	2	0
2	4	4	4	4
1	4	0	4	1
4	4	2	4	1
0	0	0	1	2

*

0	1	0
0	1	0
0	1	0

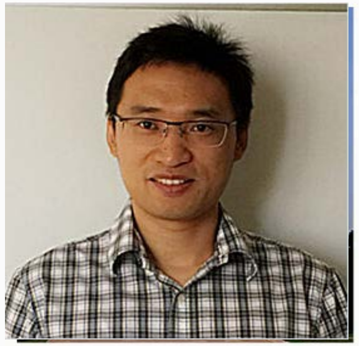
=

8	5	10
12	6	12
8	2	9

Age?

Sharpen:

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

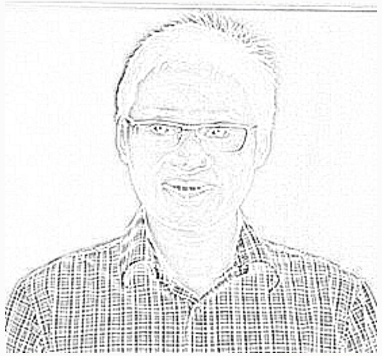


Blur:

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Edge Detect:

	0	1	0	
	1	-4	1	
	0	1	0	

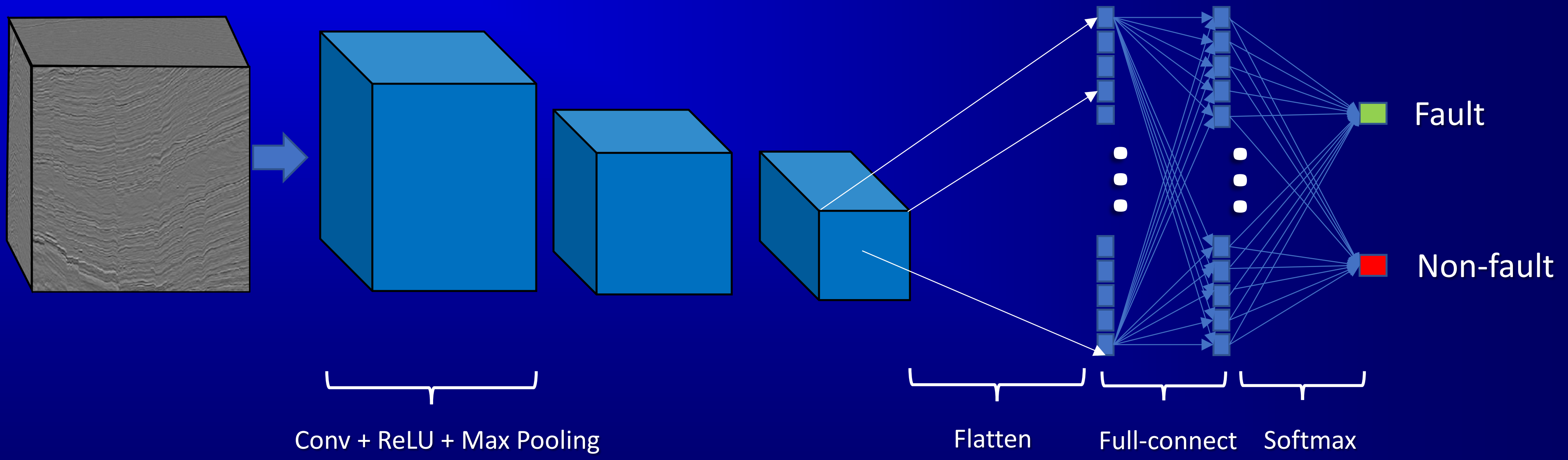


https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn

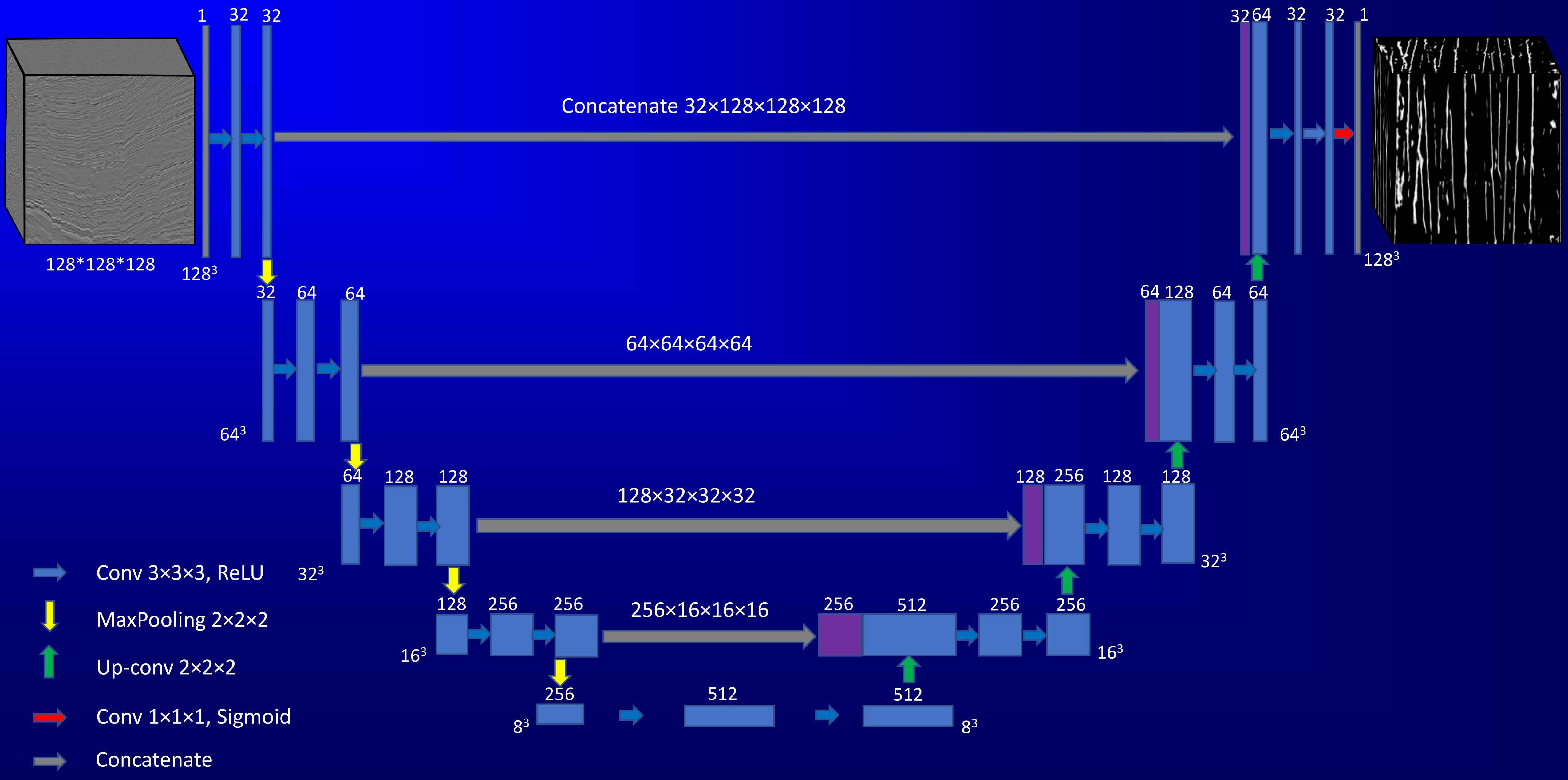
CNN workflow

- Generate training seismic images and label images
- Build a convolutional neural network model
- Train and validate the model with training seismic images and label images
- Feed the application seismic data to the model for prediction

Typical architecture of a CNN model



The U-Net

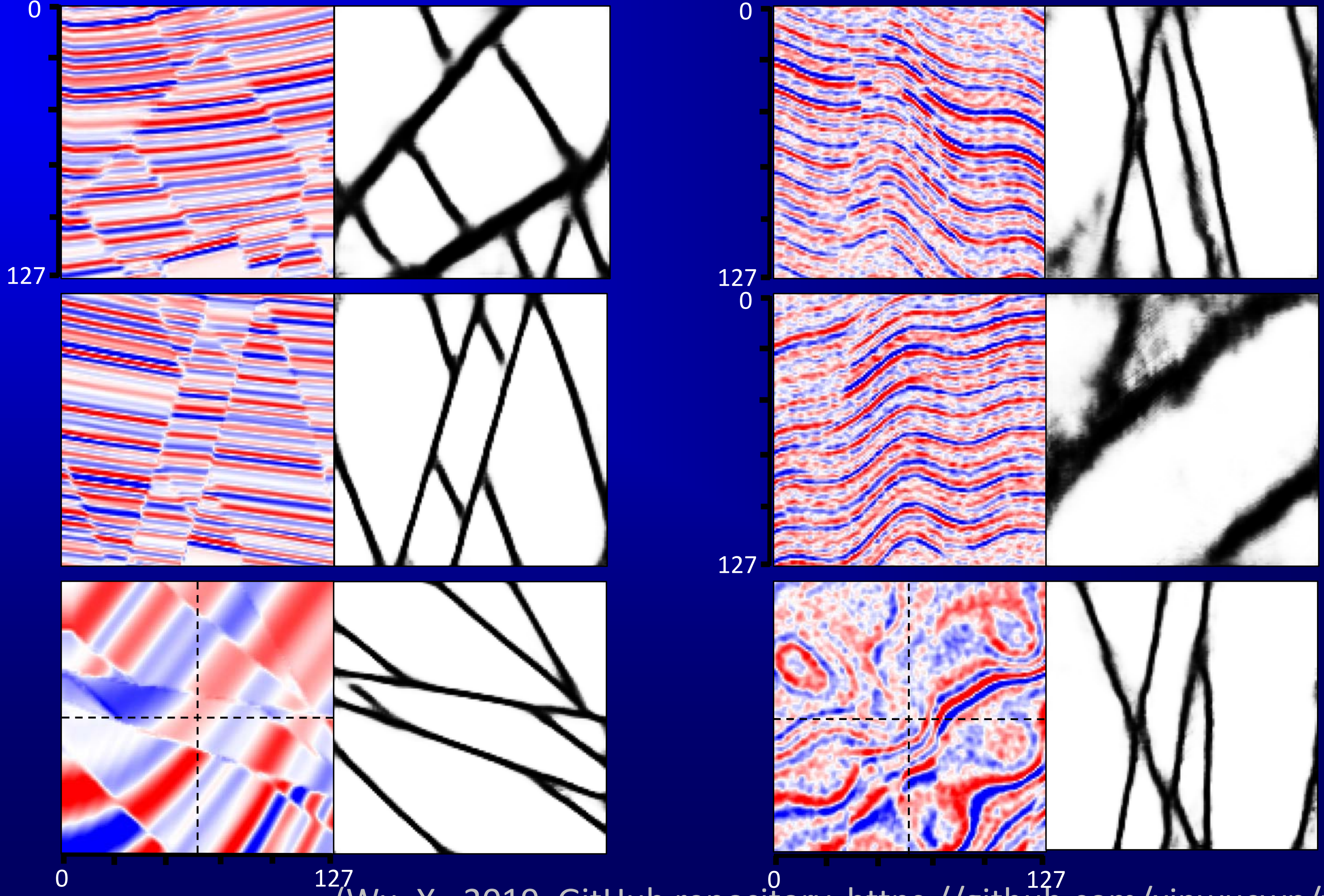


1
32
32
128*128*128
128^3
Concatenate 32×128×128×128
32
64
32
32
1
128^3
32
64
64
64^3
64×64×64×64
64
128
64
64
64^3
64
128
128
32^3
128×32×32×32
128
256
128
128
32^3
128
256
256
16^3
256×16×16×16
256
512
256
256
16^3
256
8^3
512
512
8^3
Conv 3×3×3, ReLU
MaxPooling 2×2×2
Up-conv 2×2×2
Conv 1×1×1, Sigmoid
Concatenate

U-Net architecture

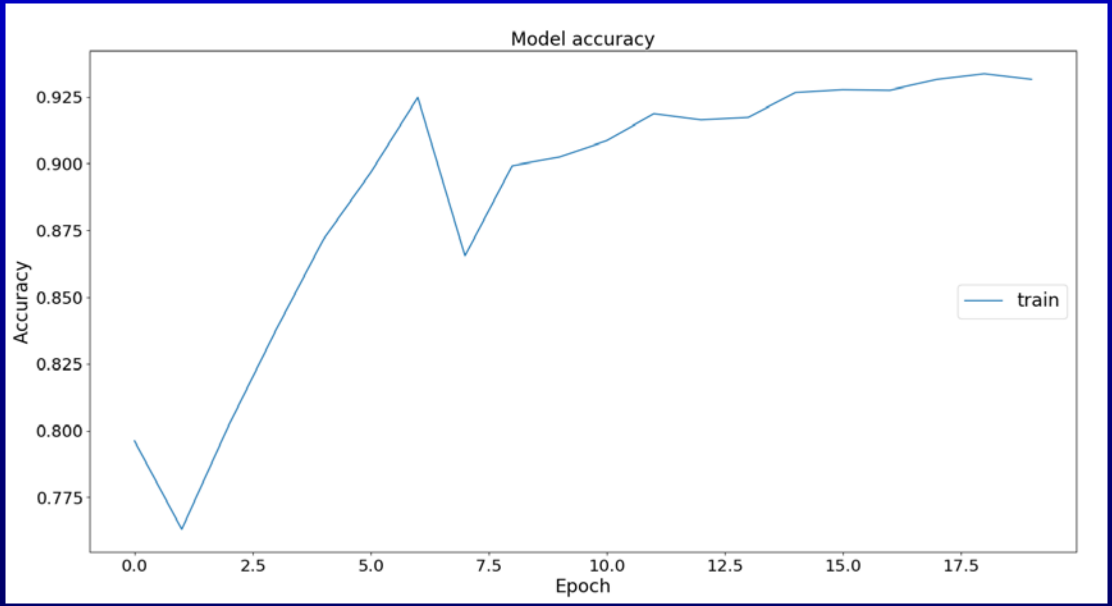
- Consists of a contracting path (capture features) and an expansive path (localization);
- A fully convolutional network without any fully connected layers;
- Increases the resolution of the output by supplying successive layers;
- Allows to input the full context;
- Needs to extrapolate the context at the border region.

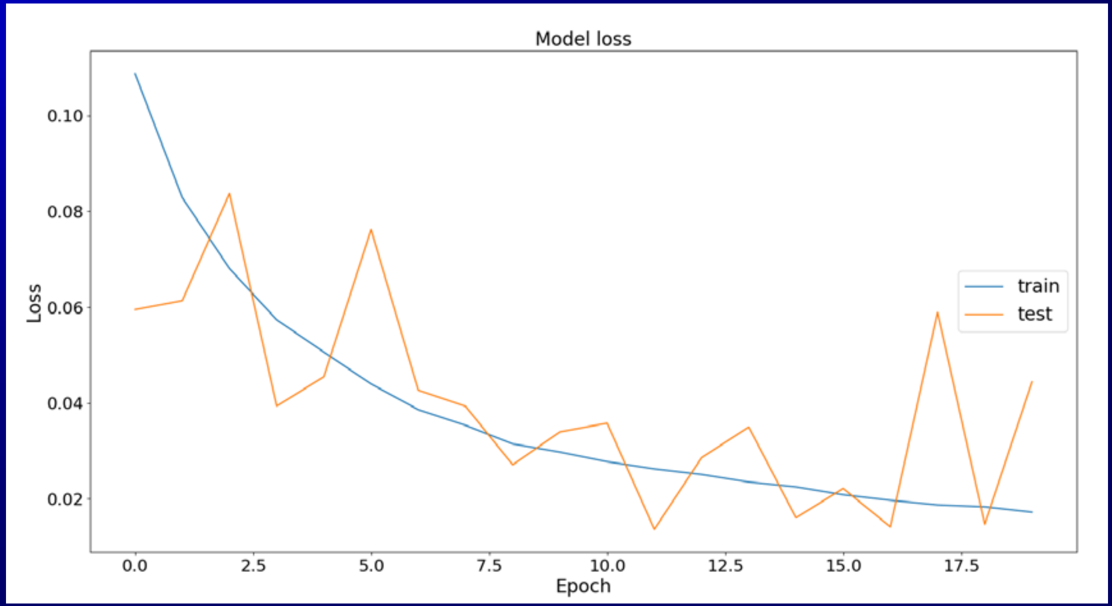
Training volumes



(Wu, X., 2019, GitHub repository, https://github.com/xinwucwp/faultSeg)

Training and validation





Application to steep normal faults (an easy problem)

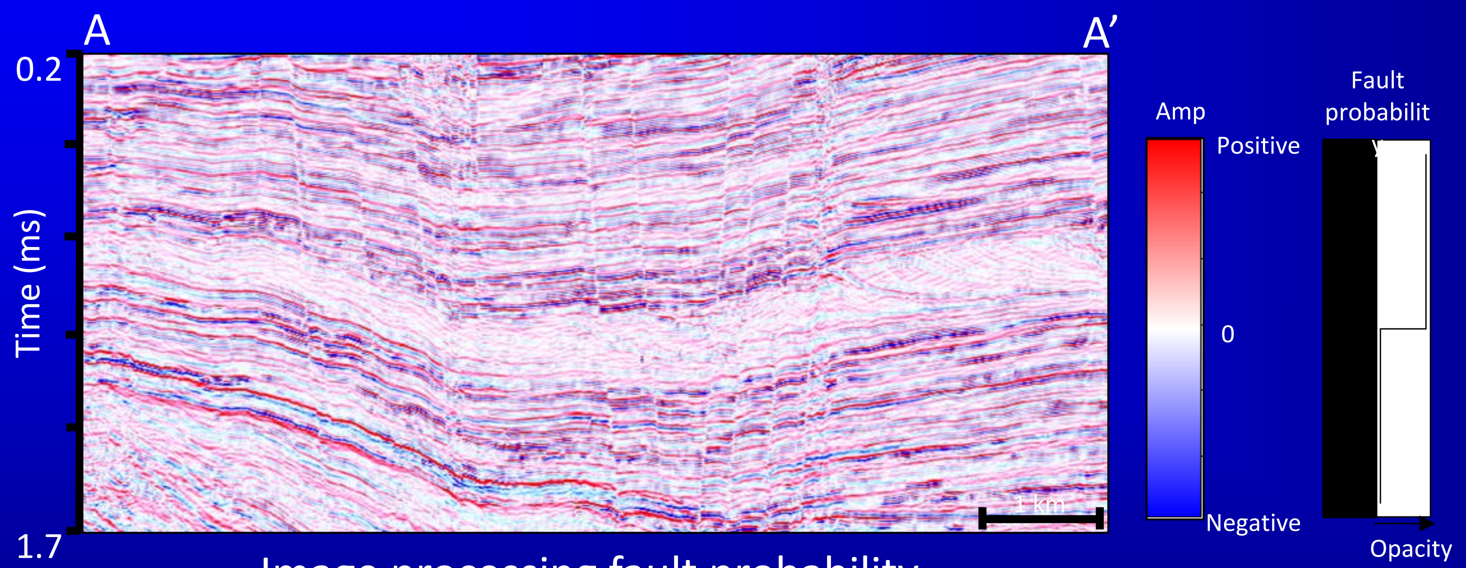
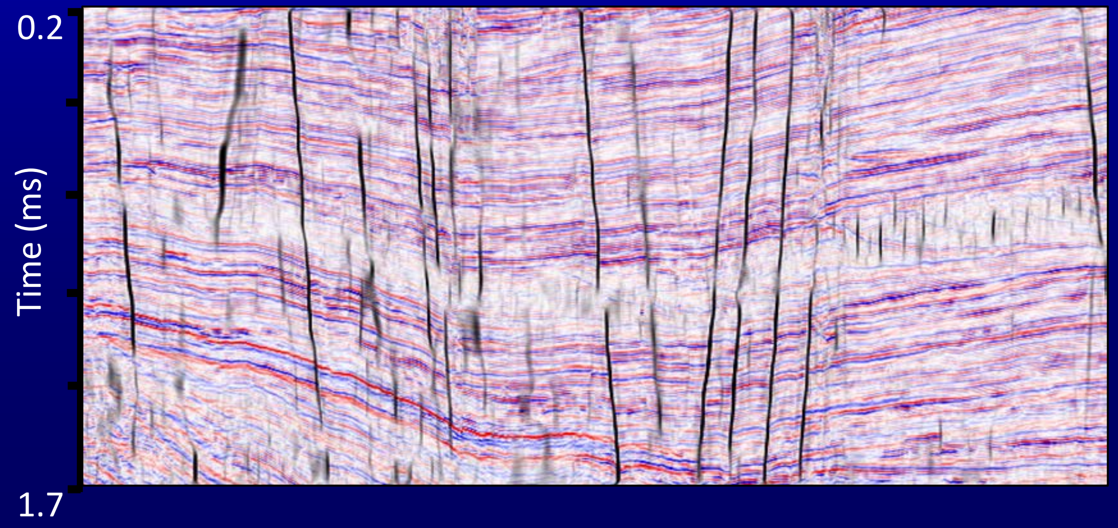
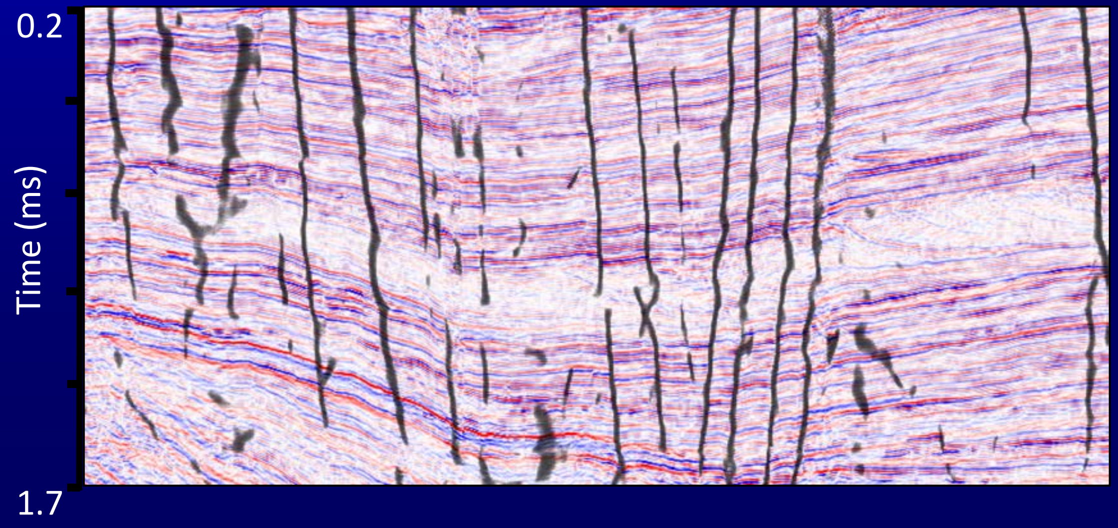


Image processing fault probability



CNN fault probability



Application to steep normal faults (an easy problem)

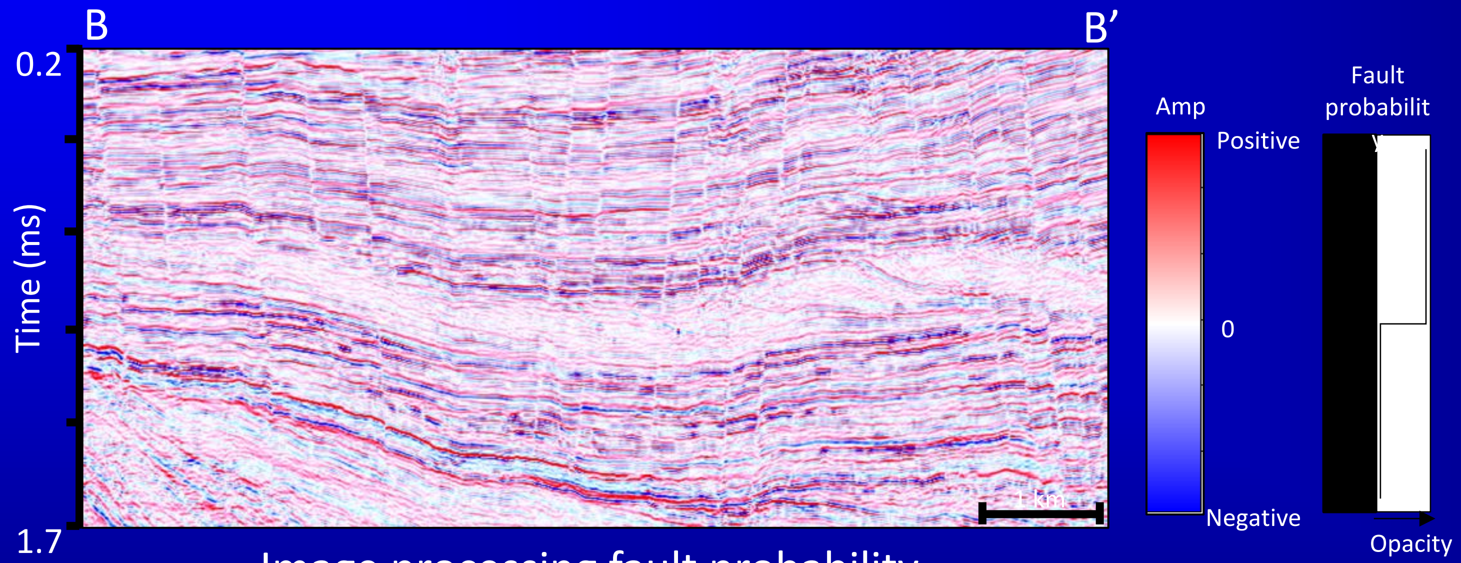
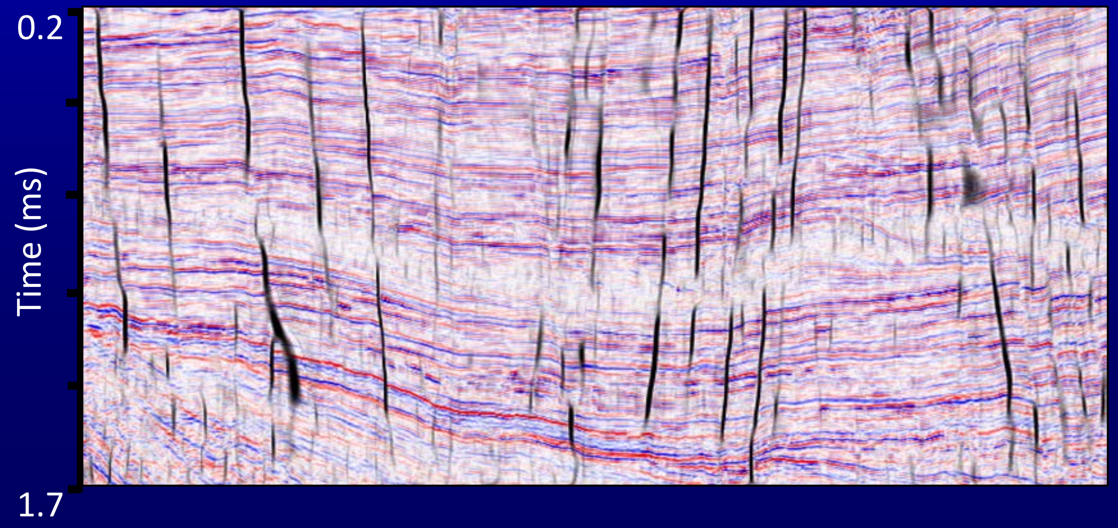
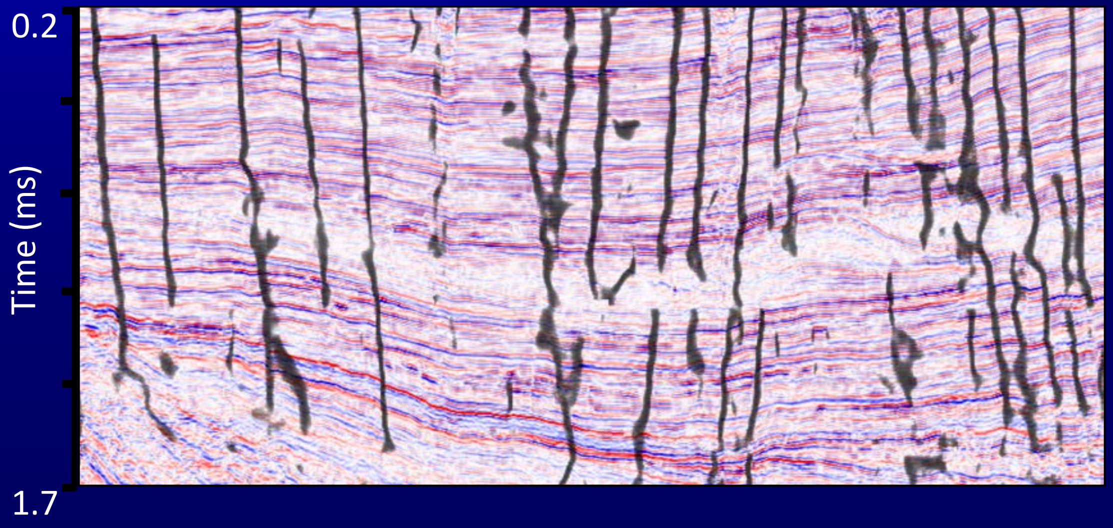


Image processing fault probability



CNN fault probability



Application to steep normal faults (an easy problem)

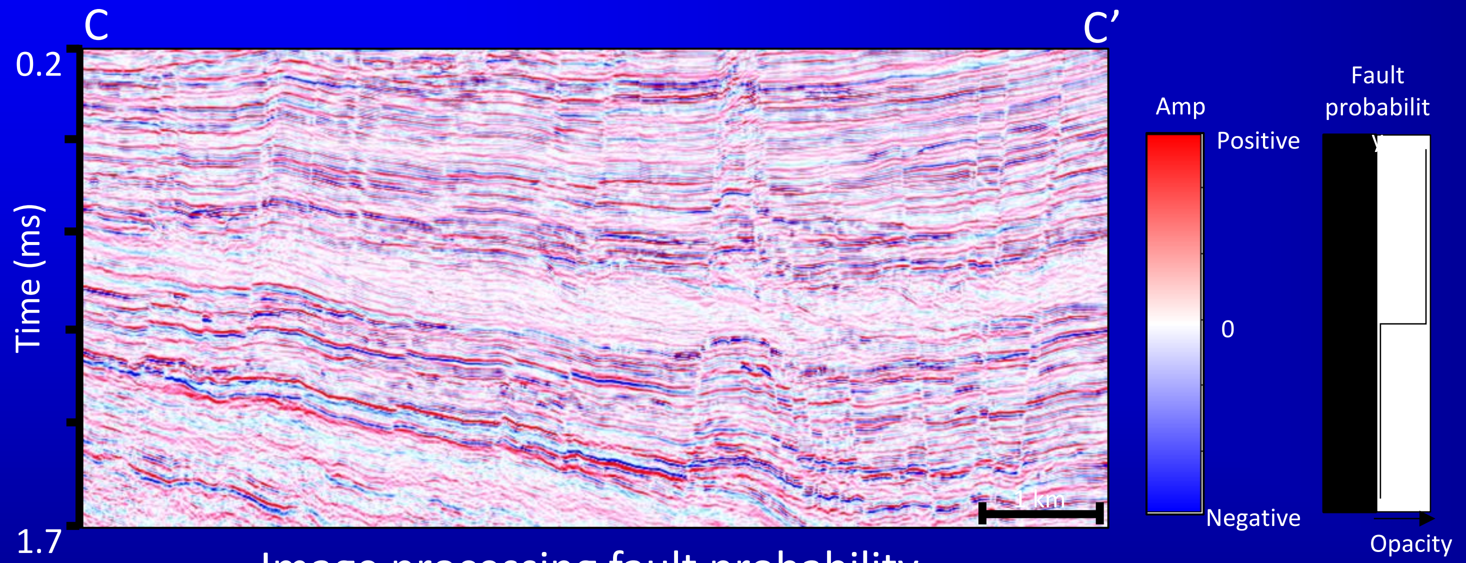
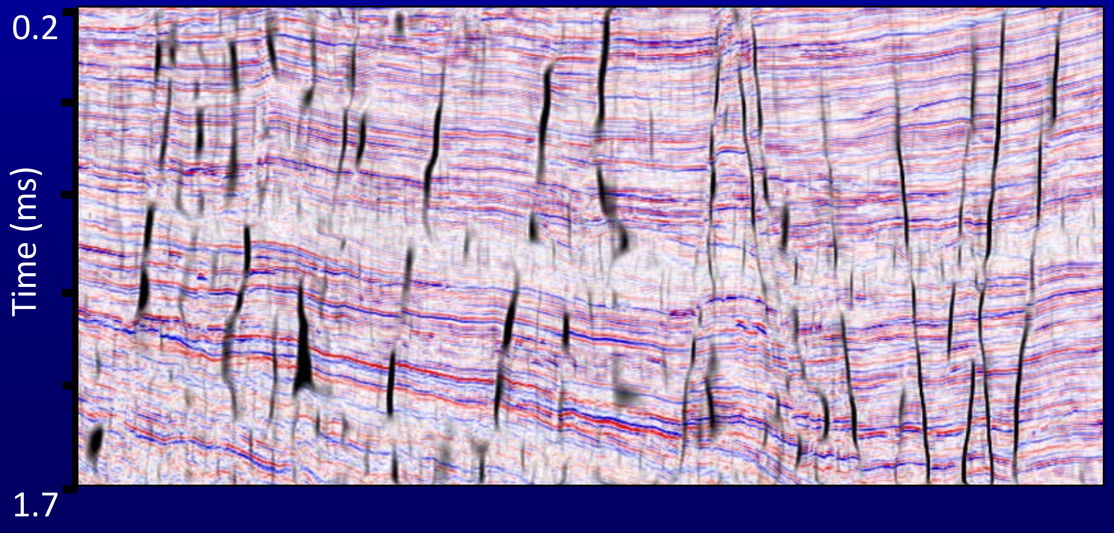
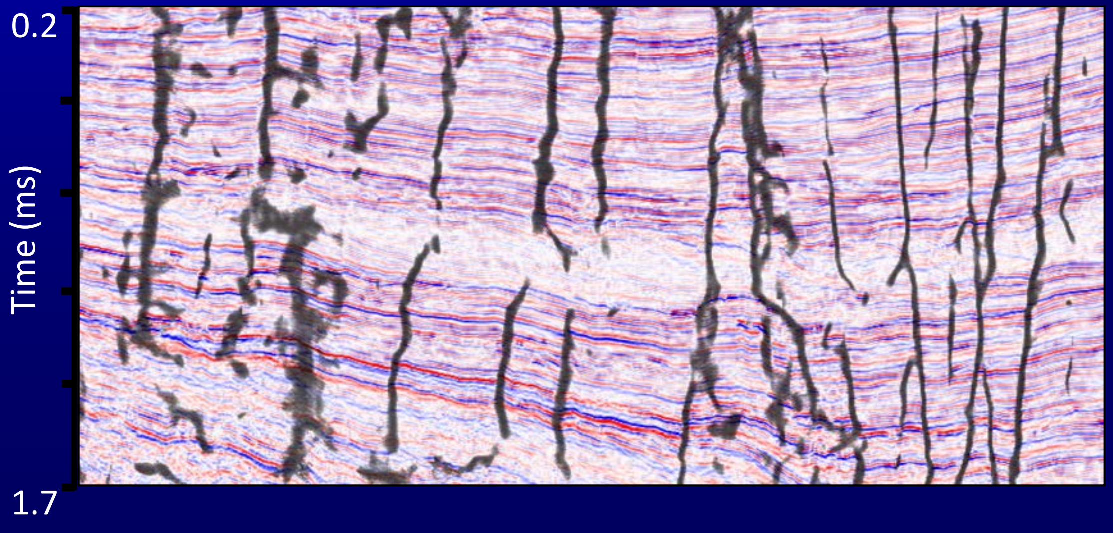


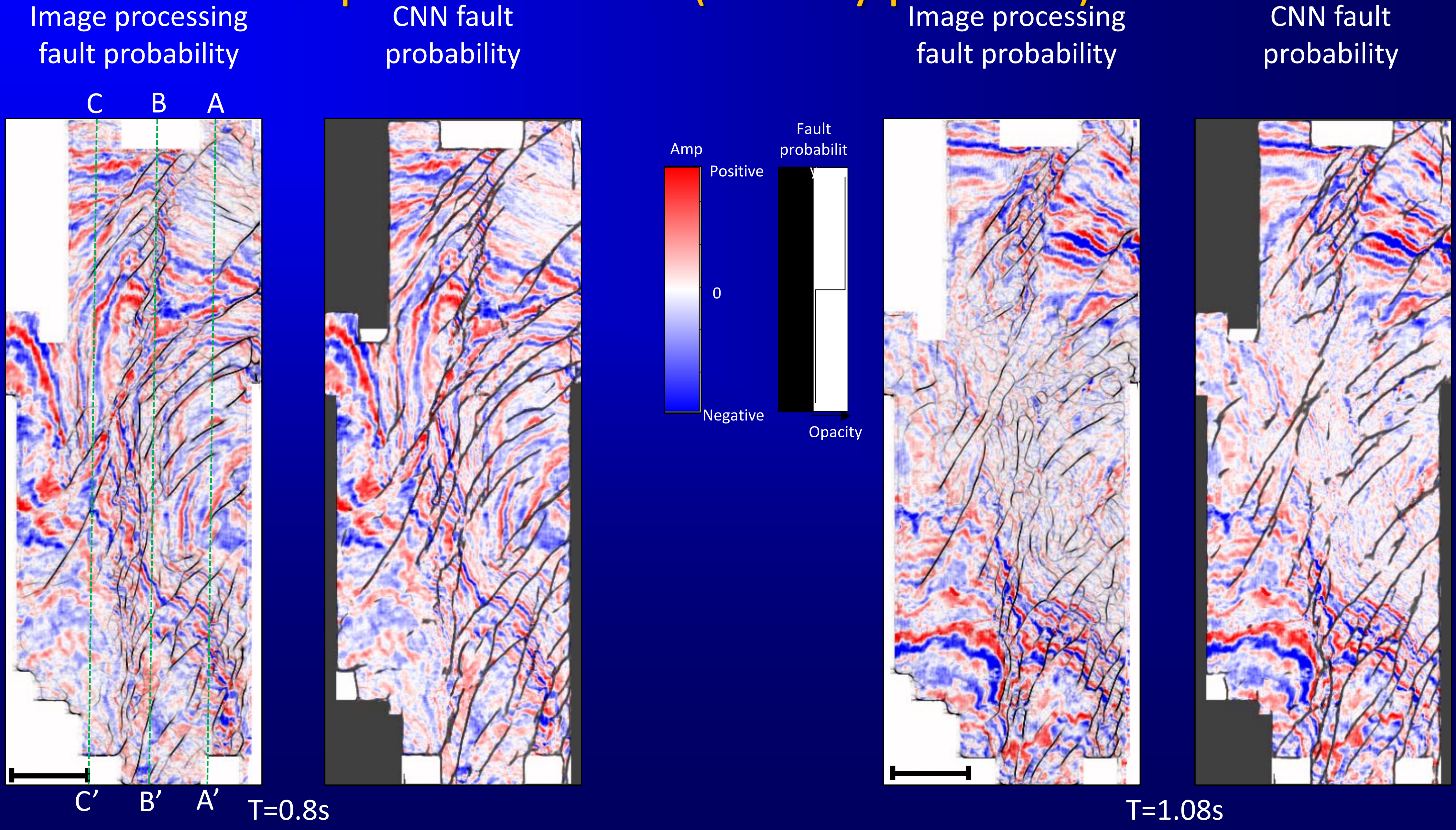
Image processing fault probability



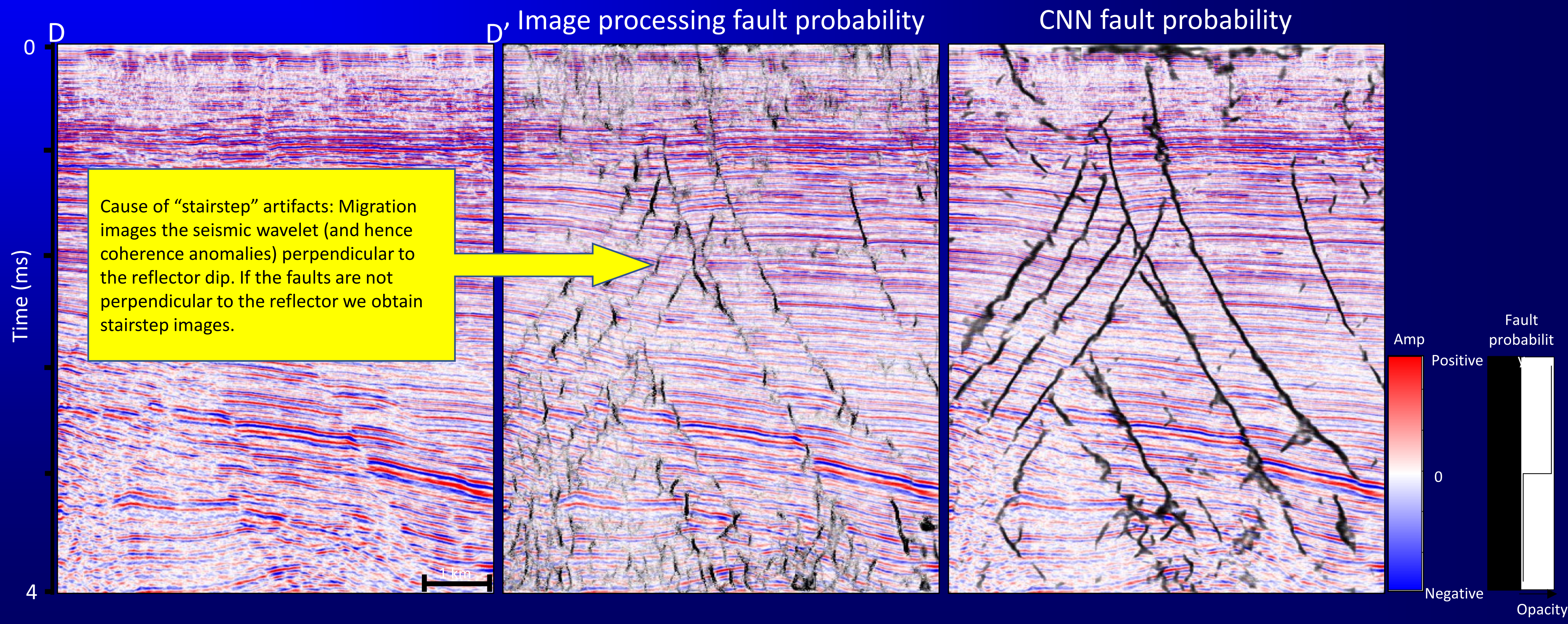
CNN fault probability



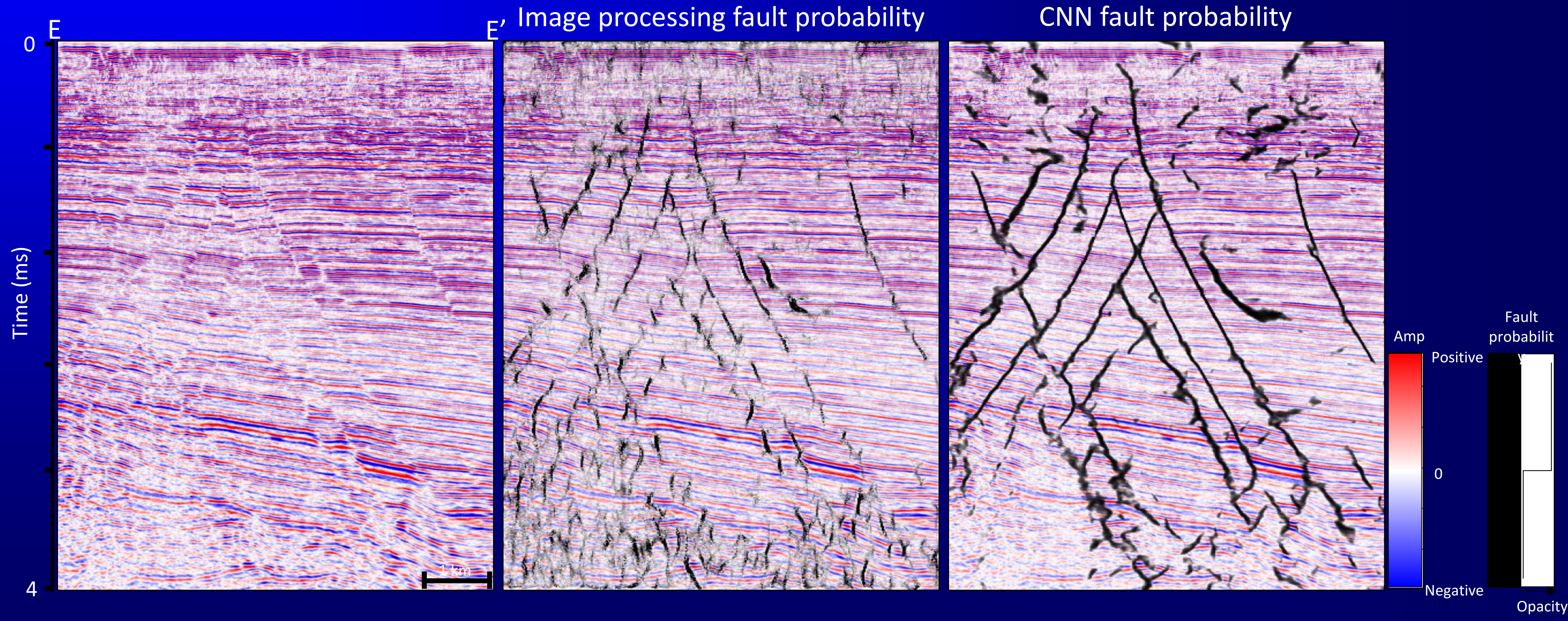
Application to steep normal faults (an easy problem)



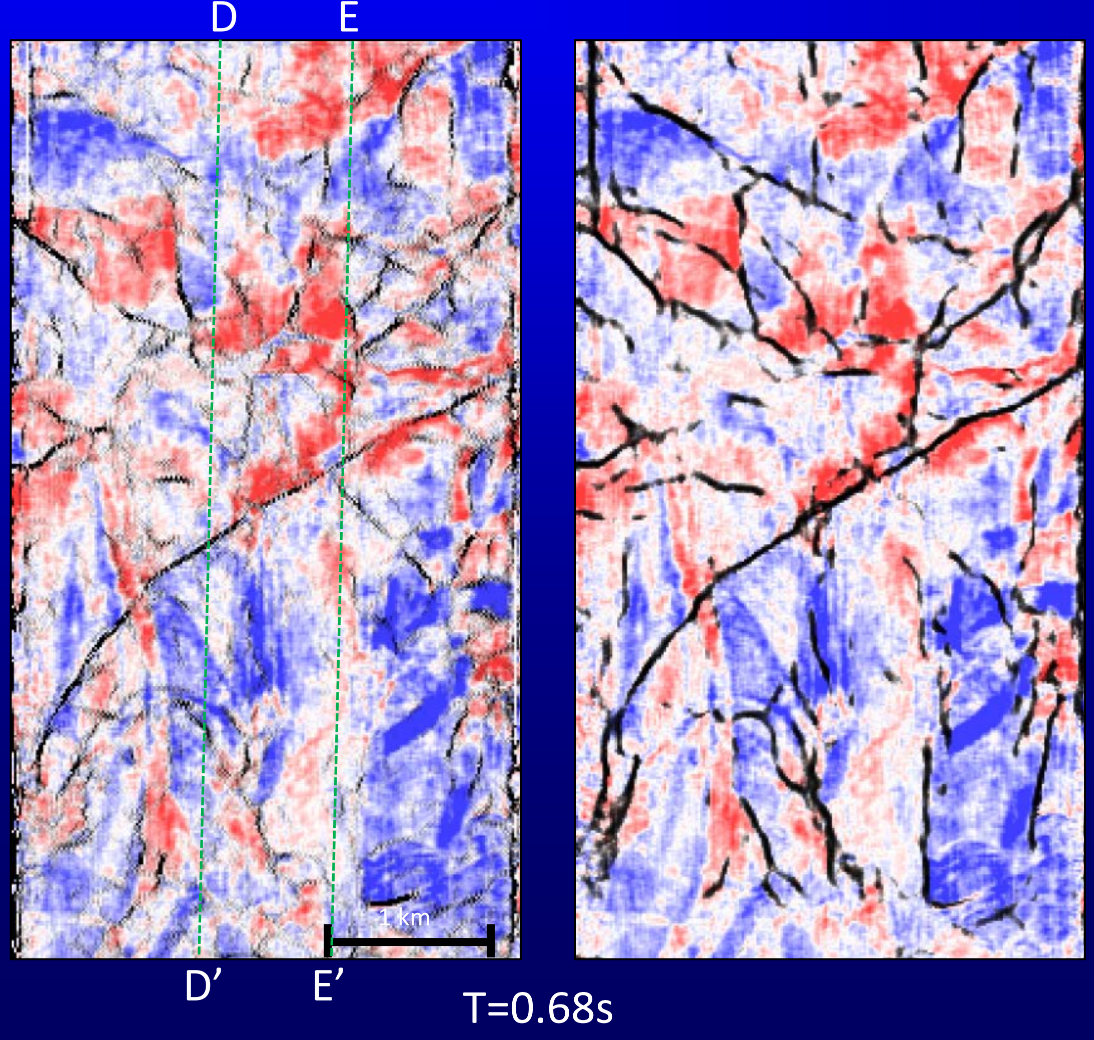
Application to moderate angle faults (an intermediate problem)

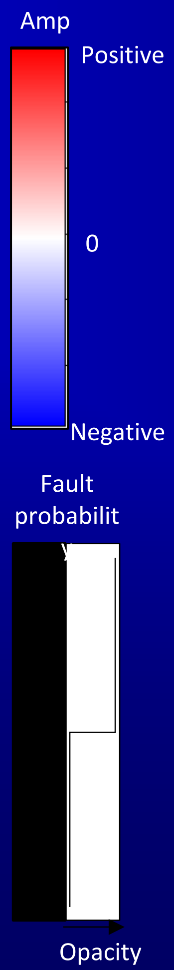


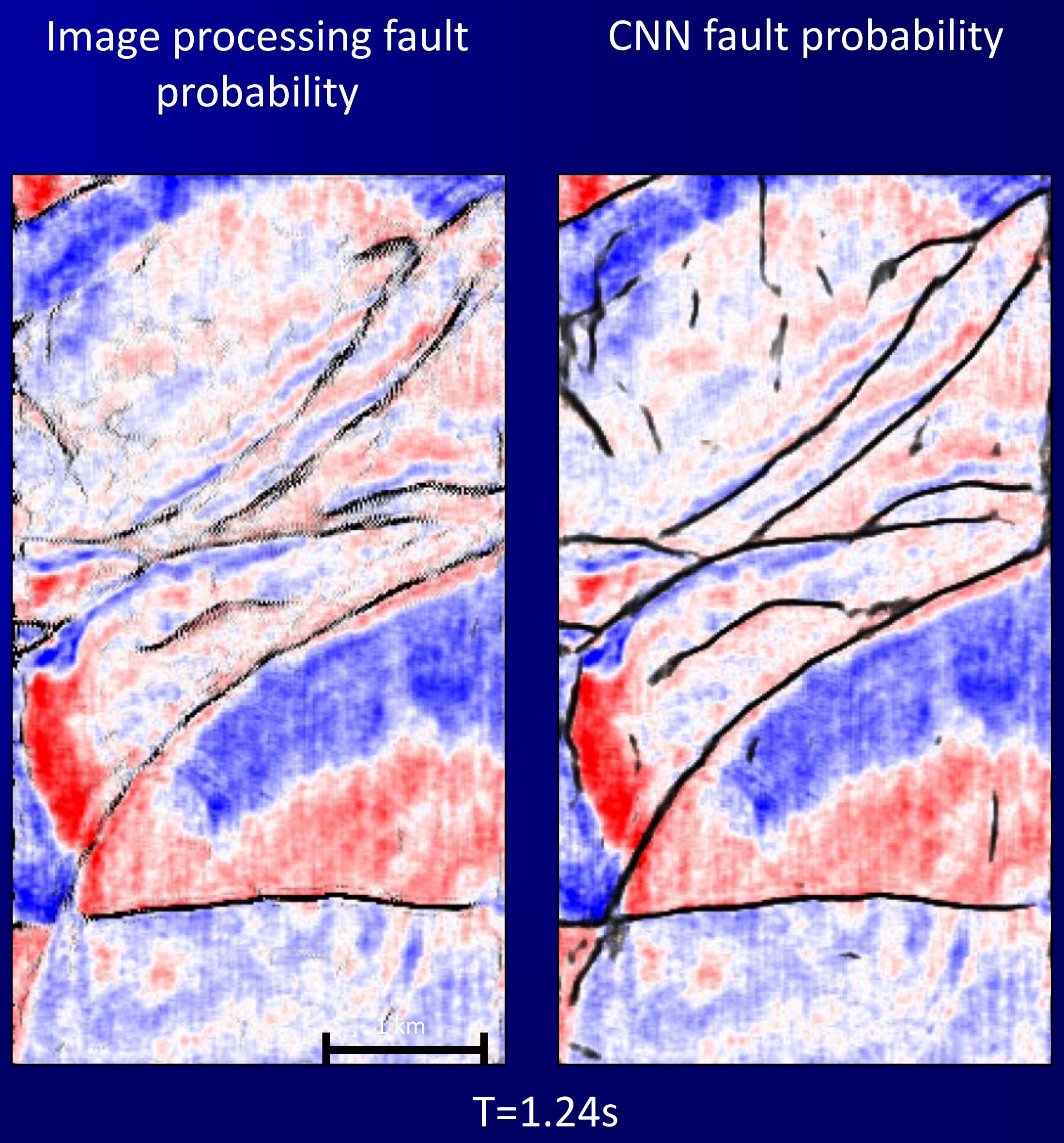
Application to moderate angle faults (an intermediate problem)



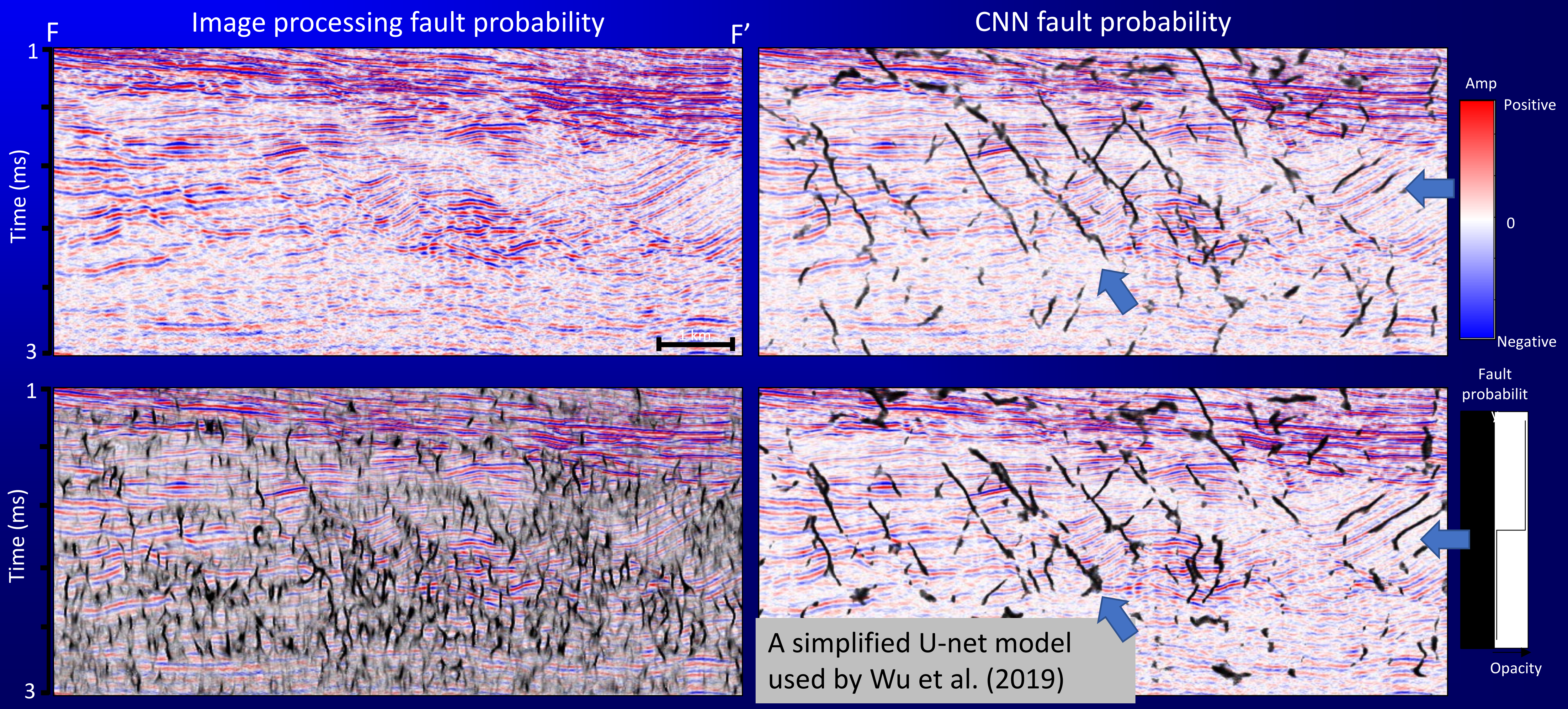
Application to moderate angle faults (an intermediate problem)



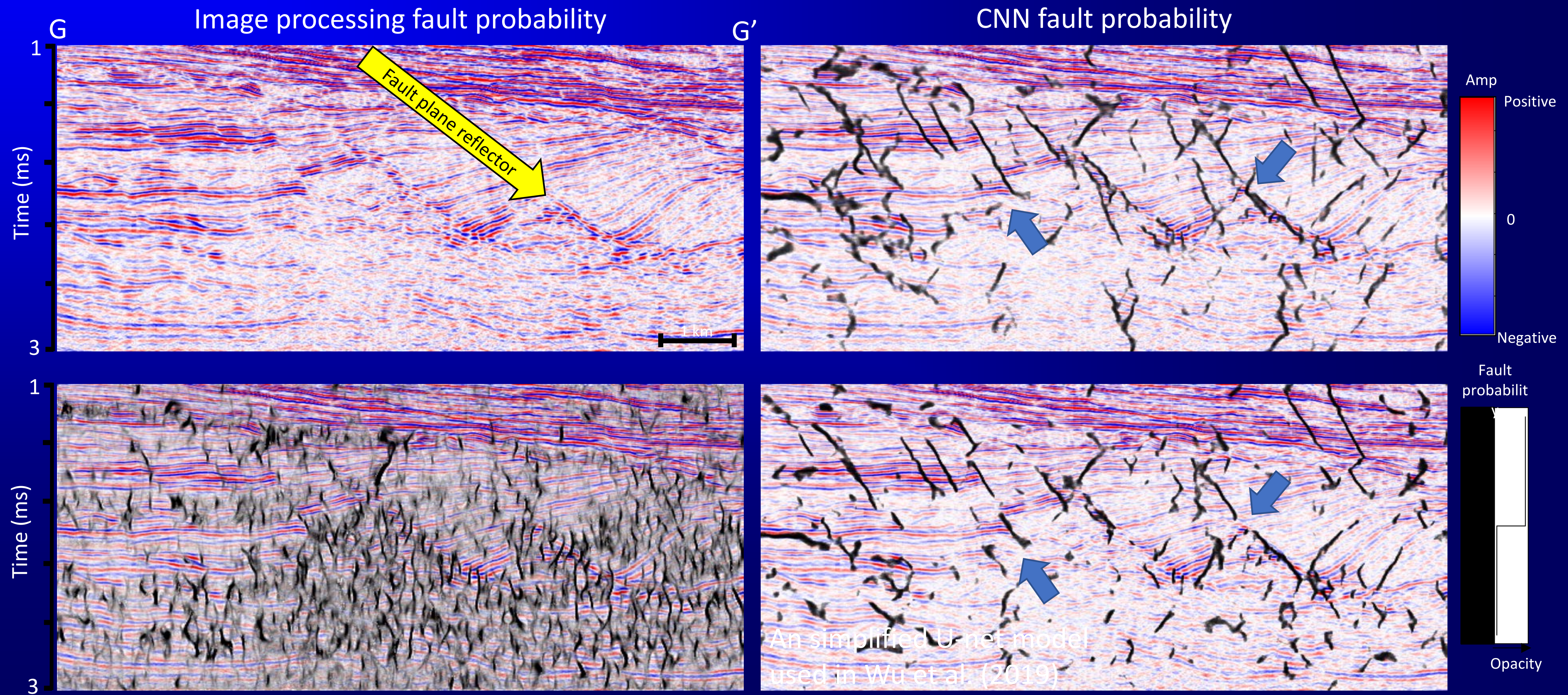




Application to listric faults (a difficult problem)



Application to listric faults (a difficult problem)



Application to listric faults (a difficult problem)

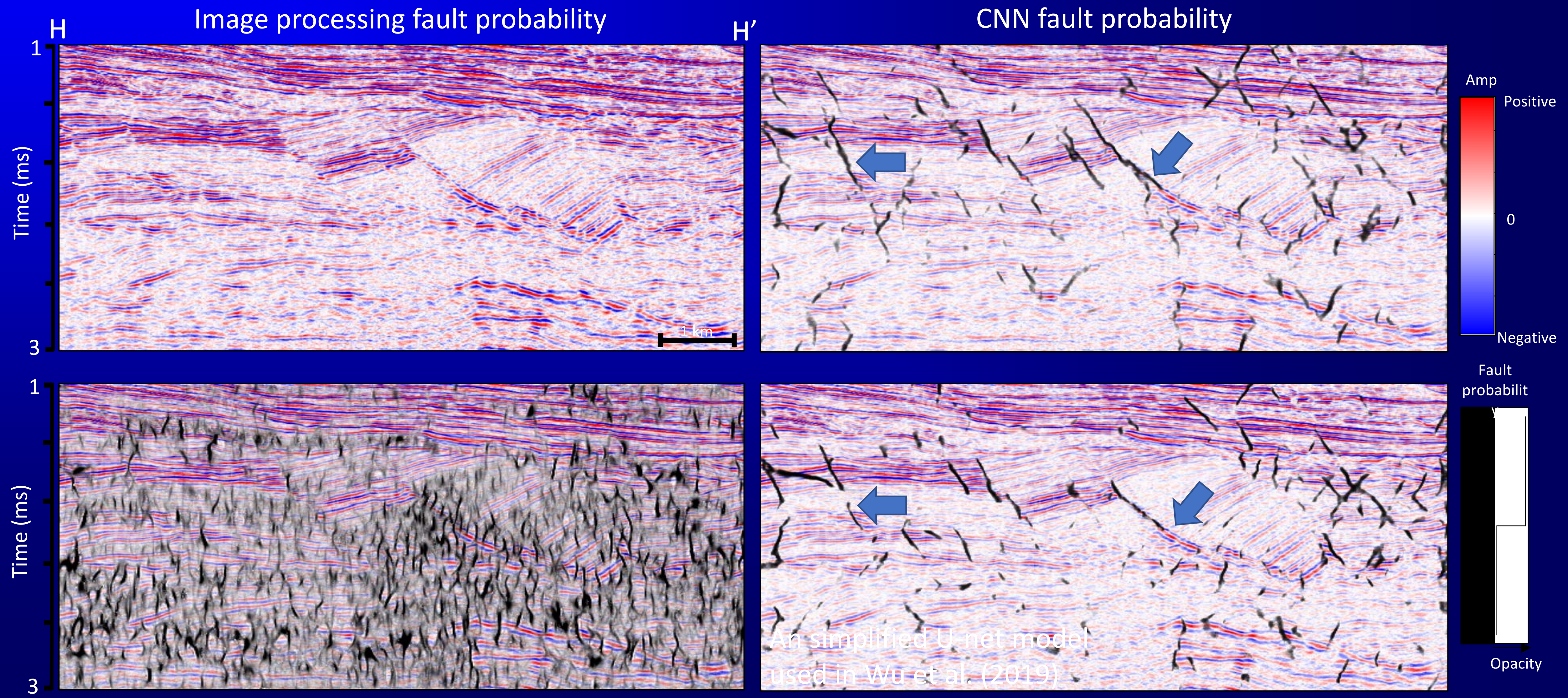
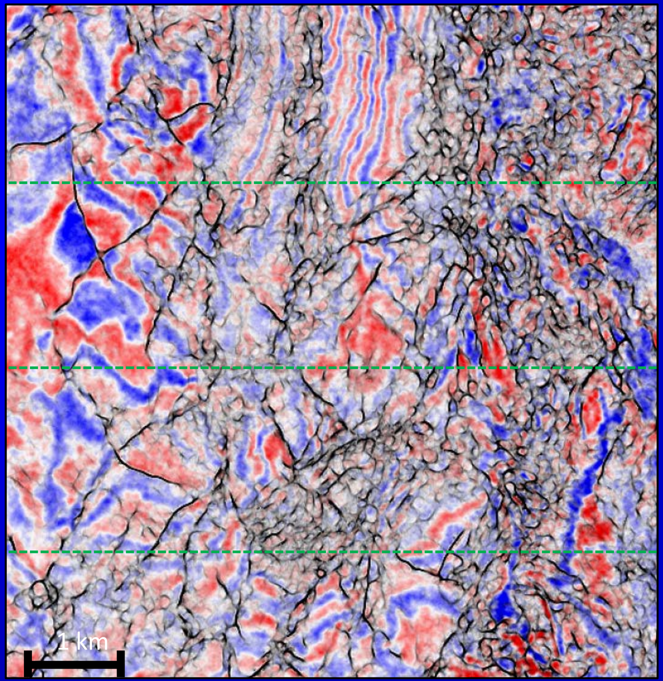


Image processing fault probability

T= 1.52s

H H'
G G'
F F'



CNN fault probability

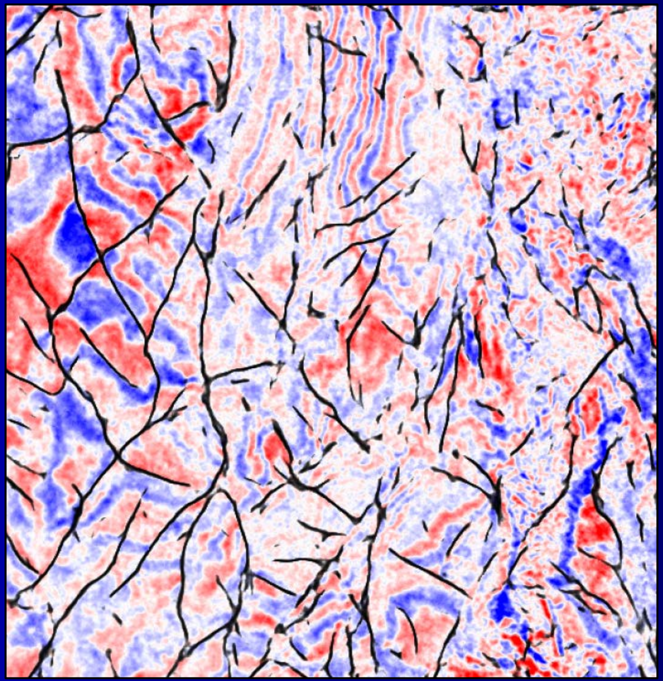
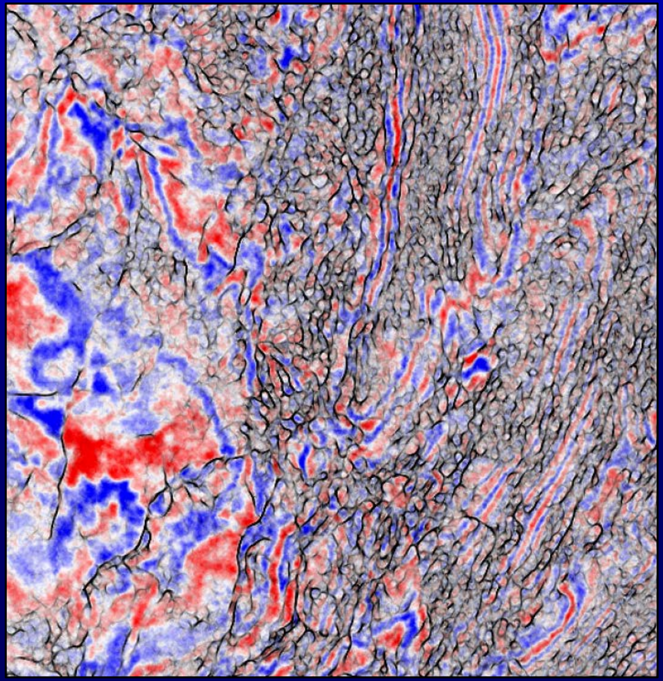
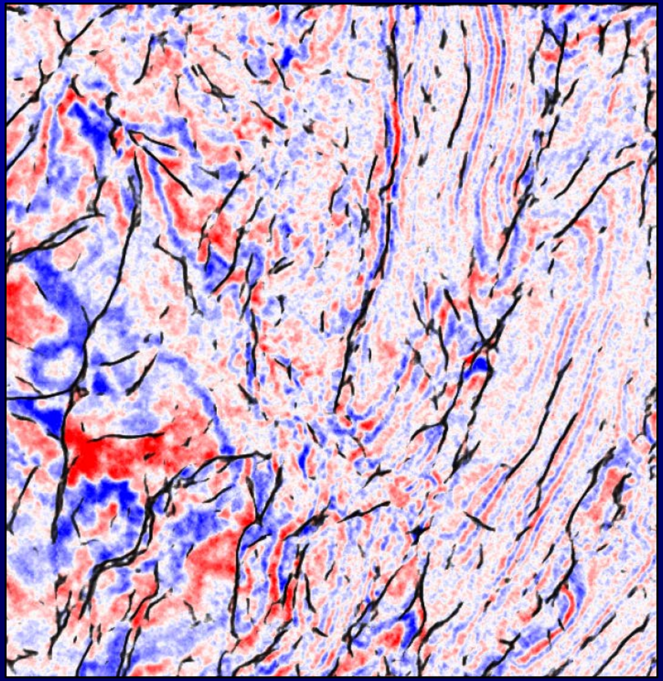


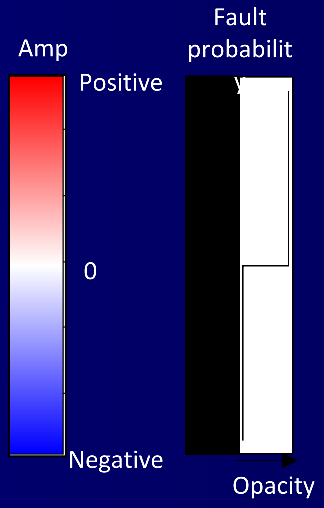
Image processing fault probability

T= 2s



CNN fault probability





Conclusions

- The U-Net architecture CNN performs well on fault detection without any human-computer interactive work;
- The CNN model trained by the synthetic training volumes can be applied to different data;
- The computation cost on training a CNN model is high, but extremely low on data prediction;
- The CNN method was trained only to be sensitive to faults, resulting in two classes – a fault, and not-a-fault;
- The CNN method does better than image processing in detecting faults with shallower dips, including those represented by fault-plane reflections;
- CNN images do not generate stair-step artifacts.

Future plans

- Generate more complex models for CNN training;
- Allow the CNN model to be trained by cropped 3D real data volumes;
- Allow interpreters to select synthetic or real data to train a new model;

References

- Guo, B., L. Li, and Y. Luo, 2018, A new method for automatic seismic fault detection using convolutional neural network: 88th Annual International Meeting, SEG, Expanded Abstracts, 1951–1955.
- Google, 2017, TensorFlow tutorials: convolutional neural networks.
- Guitton, A., 2018, 3D convolutional neural networks for fault interpretation: 80th Annual International Conference and Exhibition, EAGE, Extended Abstracts.
- Hale, D., 2013, Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3D seismic images: Geophysics, 78, no. 2, O33–O43
- Huang, L., X. Dong, and T. E. Clee, 2017, A scalable deep learning platform for identifying geologic features from seismic attributes: The Leading Edge, 36, 249–256.
- Ronneberger, O., P. Fischer, and T. Brox, 2015, U-Net: Convolutional networks for biomedical image segmentation: International Conference on Medical Image Computing and Computer-Assisted Intervention, 234–241.
- Wei, X., X. Ji, Y. Ma, Y. Wang, N. M. BenHassan, and Y. Luo, 2018, Seismic fault detection with convolutional neural network: Geophysics, 83, O97-O103.
- Wu, X., L. Liang, Y. Shi, and S. Fomel, FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation: Geophysics, 84, IM35-IM45.
- Wu, X., 2019, GitHub repository, https://github.com/xinwucwp/faultSeg, accessed March 2019
- Zhao, T., and P. Mukhopadhyay, 2018, A fault-detection workflow using deep learning and image processing: 88th Annual International Meeting, SEG, Expanded Abstracts, 1966–1970.
- Zhao, T., 2019, 3D convolutional neural networks for efficient fault detection and orientation estimation: 89th Annual International Meeting, SEG, Expanded Abstracts, 2418-2422.