Exercise objective:

To predict seismic features using the *Seismic Image to Image* workflow in the machine learning plugin. In this exercise, we will predict fault locations from seismic data.

Note: To predict real faults use the pre-trained U-Net fault predictor

In this exercise we train a U-Net to predict faults from pre-processed seismic input. The input is Edge-Preserved Smoothed (EPS) seismic data. The target is a mask volume with ones (faults) and zeros (no-faults) that was created from Thinned Fault Likelihood (TFL) computed from the EPS volume. **Note** that from a geoscientific perspective this is not necessary, since we do not need a machine learning model to predict a desired outcome that can be computed directly with an algorithm. The main purpose of this exercise is to learn how to run image-to-image workflows.







Target mask (0,1) of TFL* from EPS

*EPS and TFL-mask are **NOT** delivered with F3. To replicate this workflow first create EPS and TFL (from EPS) in the Faults & Fractures plugin. Next, create a mask from TFL with the mathematics attribute using this formula: TFL > 0.01 ? 1 : 0

Exercise objective:

Note: heavy GPU requirements

In this exercise we create 1008 cubelets of 128x128x128 samples. These cubelets are extracted from half the input - and target volumes. The trained U-Net is applied to the full volume. Application is very fast (minutes) but training takes several hours on a GPU. The graphics card used is a Nvidia GeForce with 11 GB DDR6 memory. In principle, the exercise can also be run on a CPU but then training may take several days.

Workflow:

- 1. Open the Machine Learning Control Center with the 🛼 icon.
- 2. Click on Seismic.
- **3.** Select Seismic Image Segmentation and **Press** Go.



- 4. Seismic Image Transformation window pops up.
- 5. Select Input Data in the Extract Data tab.
- 6. In the Seismic Data list, Select the EPS volume
- Specify a name for the Output Deep Learning Example Data and Press Proceed.

Tip: Additional seismic attributes can be added using checkboxes



- 8. The Deep Learning: Target Seismics Definition window pops up. Select the Fault Mask Volume
- 9. Press Proceed [Input Data Selection] >>



- 10. In the *Input Data* window Set the *Image* dimensions of the cubelets to 128 x 128 x
 128 samples. Note: to extract 2D images, set one of the dimensions to 0.
- **11. Specify** the *Inline, Crossline, Time Ranges* and the corresponding *Overlap** percentages to such that we extract approx. 1000 cubelets from one half of the input and target volumes (see image for specifications).
- **12. Specify** a name for the *Output Deep Learning Example Data* (e.g. Ex_EPS_Faults) and **Press** Extract
- **13.** When the **Extraction** is done, press Proceed



*Overlap: if the number of examples that can be extracted from a given range and overlap does not fit exactly, the last example is extracted from the boundary backwards.



- 14. Specify the *Output Deep Learning Model* name (e.g. ML_U-Net_EPS2TFL_128x128x128)
- 15. In the *Train* tab, **Select** Keras (tensorflow) as *Machine learning platform*
- 16. Select the Parameters tab

The machine learning plugin supports two platforms:

Keras (tensorflow) for deep learning (convolutional neural networks) and Pytorch. Supported models and training parameters are specified in the Parameters tab.



- 17. In the *Parameters* tab **Select** Type U-Net
- **18.** Set Batch Size to 2 A U-Net needs a lot of GPU memory in the training phase. If memory is exceeded, training stops with an error message. You can then try to rerun with a smaller batch size. Try with the largest possible batch size as training performance increases with batch size.
- **19. Set** the number of *Epochs* to 100 (this is the number of training cycles through all examples that are offered in batches of Batch Size).
- **20.** Set Early Stopping to 50. This parameter avoids early stopping when the error does not decrease after this number of Epochs.

Training Parameters Advanced Type dGB UNet Segmentation

Tip: To change the numbers in the sliders more precisely, click on the corresponding slider and use the arrow keys



21. Go back to the *Training* tab.

22. In the *Training* tab **Press** *Run*

23. The Machine Learning training log window pops up. This window can also be started by pressing the icon.
Press Reload to refresh.

Training	Parameters	Advanced	
Machine le	earning platforn	n:	
Keras (t	ensorflow)		•
			► Run

🔍 🖈 Machine Learning training log 🛛 🗸 🔨 🗸
File
118/201 [====================================
119/201 [====================================
120/201 [====================================
121/201 [=====================] - ETA: 1:46 - loss: 0.1286 - accuracy: 0.3600
122/201 [======================] - ETA: 1:44 - loss: 0.1282 - accuracy: 0.3647
123/201 [======================] - ETA: 1:43 - loss: 0.1283 - accuracy: 0.3689
124/201 [=========================] - ETA: 1:41 - loss: 0.1280 - accuracy: 0.3734
125/201 [=========================] - ETA: 1:40 - loss: 0.1280 - accuracy: 0.3775
126/201 [========================] - ETA: 1:39 - loss: 0.1280 - accuracy: 0.3816
127/201 [=======================] - ETA: 1:37 - loss: 0.1280 - accuracy: 0.3856
128/201 [=====================] - ETA: 1:36 - loss: 0.1279 - accuracy: 0.3897
129/201 [=====================] - ETA: 1:35 - loss: 0.1277 - accuracy: 0.3937
130/201 [======================] - ETA: 1:33 - loss: 0.1276 - accuracy: 0.3977
131/201 [======================] - ETA: 1:32 - loss: 0.1276 - accuracy: 0.4014
132/201 [=====================] - ETA: 1:31 - loss: 0.1273 - accuracy: 0.4053
133/201 [======================] - ETA: 1:29 - loss: 0.1273 - accuracy: 0.4090
134/201 [==============================] - ETA: 1:28 - loss: 0.1272 - accuracy: 0.4127
135/201 [====================================
136/201 [====================================
137/201 [===============================] - ETA: 1:24 - loss: 0.1268 - accuracy: 0.4235
138/201 [==============================] - ETA: 1:22 - loss: 0.1268 - accuracy: 0.4269
139/201 [========================] - ETA: 1:21 - loss: 0.1267 - accuracy: 0.4303
140/201 [========================] - ETA: 1:20 - loss: 0.1267 - accuracy: 0.4337
141/201 [====================================
142/201 [====================================
143/201 [====================================
Close Cose

Workflow cont'd:	
24. When training is finished press ⊘ Proceed≫ or select the <i>Apply</i> tab	>
25. Select the trained model ML_U- Net_EPS2TFL_128x128x128 and press Proceed.	 Seismic Image Segmentation - 3D Extract in Train & Apply Filter Mcdel ML_2DUnet_Seis2Salt_0x128x128 ML_JOUnet_Faults_128x128x128 ML_U-Net_EPS2TFL_128x128x128 MLU-Net_EPS2TFL_128x128x128 MLU-Net_EPS2TFL_128x128x128 Type: Selsmic Image Transformation Data Type: Cube Number of inputs: 1 Number of inputs: 1 Number: Fault_Mask Created by: krist Created by: krist Created by: krist Created by: krist Created 18: Wed 31 May 2023, 10:51:12 Object ID: 100060.85
	Proceed OClose O

- 26. In the *Apply* window **Select** the *Input Cube* Edge_Preserved_Smoothed.
- 27. Specify the *Output Cube* name that will be created by the trained model, e.g. ML_U-Net_TFL_prediction.
- **28. Press** Run to start processing.

apply 'ML_U-Net_EPS2TFL_128x128x128'	—		×
Input for 'EPS' 👺 EPS 🗸 🕒 Select			
Apply Overlap (%) inl: 10 🗘 crl: 10 🌩 z: 10 🗘 Merge Mode Blend	~		
Volume subselection 100/300-750/1250 (463 samples)			
Output 'Fault_Mask' ML_UNet_Faults ~ Select (*	CBVS		~
Predict using • GPU CPU			
Execute in Batch Options			
Ru	in 🔞 🤇	Close	0

29. A *Progress Viewer* window pops up. Applying the trained U-Net is very fast. The resulting fault prediction can be viewed e.g. as overlay on the EPS of inline 425.



Processing finished successfully



Inline 500 EPS + TFL mask



Inline 500 EPS + U-Net Prediction

 \times