Localizing microseismic events on field data using a U-Net based convolutional neural network trained on synthetic data

Vinard, N. A.; Drijkoningen, G. G.; Verschuur, D. J.

DOI
10.1190/geo2020-0868.1

Publication date
2021

Document Version
Accepted author manuscript

Published in
Geophysics

GEOPHYSICS®

Localizing microseismic events on field data using a U-Net based convolutional neural network trained on synthetic data

Journal:	*Geophysics*
Manuscript ID	GEO-2020-0868.R3
Manuscript Type:	Technical Paper
Keywords:	microseismic, hydraulic fracturing, monitoring, neural networks, machine learning
Manuscript Focus Area:	Passive Seismic and Microseismic Methods

DOI:10.1190/geo2020-0868.1

DOI: 10.1190/geo2020-0668.1

Localizing microseismic events on field data using a U-Net based convolutional neural network trained on synthetic data

N. A. Vinard*, G. G. Drijkoningen* and D. J. Verschuur†

**Delft University of Technology,*

Department of Geoscience and Engineering,

Stevinweg 1, 2628CN Delft,

The Netherlands

†*Delft University of Technology,*

Department of Imaging Physics,

Stevinweg 1, 2628CN Delft,

The Netherlands

(November 6, 2021)

Running head: **U-Net based microseismic localization**

DOI: 10.1190/geo2020-0806.1

Localizing microseismic events on field data using a U-Net based convolutional neural network trained on synthetic data

(November 6, 2021)

Running head: **U-Net based microseismic localization**

ABSTRACT

Hydraulic fracturing plays an important role when it comes to the extraction of resources in unconventional reservoirs. The microseismic activity arising during hydraulic fracturing operations needs to be monitored to both improve productivity and to make decisions about mitigation measures. Recently, deep learning methods have been investigated to localize earthquakes given field-data waveforms as input. For optimal results, these methods require large field data sets that cover the entire region of interest. In practice, such data sets are often scarce. To overcome this shortcoming, we propose initially to use a (large) synthetic data set with full waveforms to train a U-Net that reconstructs the source location as a 3D Gaussian distribution. As field data set for our study we use data recorded during hydraulic fracturing operations in Texas. Synthetic waveforms were modelled using a velocity model from the site that was also used for a conventional diffraction-stacking (DS) approach. To increase the U-Nets' ability to localize seismic events, we augmented the synthetic data with different techniques, including the addition of field noise. We select the

DOI: 10.1190/geo2020-0863.1

best performing U-Net using 22 events that have previously been identified to be confidently localized by DS and apply that U-Net to all 1245 events. We compare our predicted locations to DS and the DS locations refined by a relative location (DSRL) method. The U-Net based locations are better constrained in depth compared to DS and the mean hypocenter difference with respect to DSRL locations is 163 meters. This shows potential for the use of synthetic data to complement or replace field data for training. Furthermore, after training, the method returns the source locations in near real-time given the full waveforms, alleviating the need to pick arrival times.

INTRODUCTION

Many of our society's energy requirements are provided by oil and gas that are extracted from conventional and unconventional reservoirs in the subsurface. In certain unconventional reservoirs, hydraulic fracturing (HF) is a prerequisite step for the extraction of the resources. During HF operations fluids are injected into the subsurface at high pressures to fracture the surrounding rocks (Li et al., 2020). Naturally, this leads to seismicity that is generally weak with moment magnitudes (Mw) around 0, typically referred to as microseismicity (Van Der Baan et al., 2013).

Microseismic monitoring has been around for many decades and finds applications in different industries such as the mining industry to gain insights about the rock-mass response to mining activities (Mendecki, 1993), the hydroelectric power industry to monitor the seismicity induced by water reservoirs (Simpson et al., 1988) and the geothermal industry (Pearson, 1981). However, herein we focus on hydraulic fracturing.

Continuous microseismic monitoring in a hydraulic fracturing setting provides important information to both optimize production (Maxwell et al., 2002) and to decide about mitigation measures to prevent larger events (Kao et al., 2018). Some key tasks of microseismic monitoring include event detection, source-location identification, event-magnitude evaluation and source-mechanism inversion (Li et al., 2019). The location of events is important for several reasons. First, it helps to differentiate between events linked to the current anthropogenic activity and other types of events. Second, the event locations provide a map of the created fractures. Finally,

DOI: 10.1190/geo2020-0869.1

the locations provide a starting point for understanding the source mechanism.

In some situations HF is carried out in multiple stages lasting over extended time periods. For example the operations within the Duvernay Formation in Canada were carried out over 153 stages in two time periods, the first one lasting 17 and the second one 16 days (McKean et al., 2019; Rodríguez-Pradilla and Eaton, 2020). In another example 19 stages each lasting about 3 hours were performed between the 28 October and 10 November 2014 in the southern Sichuan Basin, China (Chen et al., 2018). Similarly, the microseismic data used in this study, monitored HF activities over a period of 3 weeks in the Barnett Shale in Texas, USA, using a near-surface permanent installation consisting of vertical component geophones (Kratz et al., 2012). In such situations, where a lot of data needs to be processed, it may be desirable to have a method that retrieves the event locations in near real-time either while the operations are ongoing or in a post-processing step.

Waveform-based source-localization methods are commonly used in the field of microseismic monitoring (Li et al., 2020). One such method involves stacking the wavefields at different stations along expected travel-time functions to improve the signal-to-noise ratio (S/N) (Anikiev et al., 2014; Trojanowski and Eisner, 2017). Since the source location and origin time are unknown, a grid search over these variables is required to correctly stack the waveforms. Due to the source mechanism of the events, different polarities can be observed at different stations. Therefore, an additional grid search to align the polarities at the different stations can be used, which can improve detection (Chambers et al., 2014; Anikiev et al., 2014).

DOI: 10.1190/geo2020-0863.1

Machine learning and in particular deep-learning techniques have been widely applied in seismology for a wide number of tasks such as seismic phase detection and picking (Zhu et al., 2019; Zhu and Beroza, 2019; Ross et al., 2018b; Zhang et al., 2020a), P-wave picking and first-motion polarity determination (Ross et al., 2018a), microseismic monitoring in mining (Mousavi et al., 2016; Huang et al., 2018; Johnson et al., 2021), earthquake early warning (Jozinović et al., 2020; Münchmeyer et al., 2021), earthquake signal detection (Perol et al., 2018; Mousavi et al., 2019) and earthquake magnitude estimation (Lomax et al., 2019; Mousavi and Beroza, 2020a). Convolutional neural networks (CNNs) have also been applied in recent years to identify the location of seismic events. Kriegerowski et al. (2019) formulate the source localization problem as a regression task returning a (x,y,z)-location for the event. Furthermore, they used the waveforms from multiple three-component stations as input data to their CNN. Mousavi and Beroza (2020b) present a Bayesian deep learning approach to identify the location of global earthquakes from single-station observations. van den Ende and Ampuero (2020) propose a location algorithm that includes the information of the receiver locations using a graph-based deep learning approach. Zhang et al. (2020b) use a fully convolutional network (FCN) (Long et al., 2015) with multiple up-sampling layers to retrieve the source location of induced events in Oklahoma. They used the FCN to return the source location as a 3D probability density function where the peak is defined at the correct source location. In all these recent works (Kriegerowski et al., 2019; Zhang et al., 2020b; Mousavi and Beroza, 2020b; van den Ende and Ampuero, 2020), the models were trained and applied to field data. In some of those publications it was suggested that synthetic data could be used to

augment their data sets, especially to fill existing gaps in the field data. Synthetic waveforms have been used in the past to train neural networks to invert for source parameters of natural earthquakes (Käufl et al., 2016a,b). The neural networks used in these works were trained to return the posterior probability density functions for the focal depth, longitude and latitude of the earthquakes as well as other source parameters given the seismic waveforms as input. This was achieved by using Mixture Density Networks (Bishop, 1994, 1995).

In this work we model synthetic seismic data given a reasonable velocity model of the area of interest to generate a large labeled data set. The label associated to each synthetic waveform is represented as a 3D Gaussian distribution where the peak of the distribution is defined at the source location, as proposed by Zhang et al. (2020b). The data consist of seismic waveforms from multiple stations, where only the vertical component is used. This synthetic data set is used to train a slightly modified version of a U-Net, a supervised deep learning algorithm originally developed for biomedical image segmentation (Ronneberger et al., 2015). After training we apply the U-Net to microseismic field data recorded during hydraulic-fracturing operations in Texas, USA. During training we apply different types of data augmentations including the addition of field noise. To validate the U-Net we first apply it to a smaller number of events, so-called master events, that were defined by Alexandrov et al. (2020). The U-Net that performs best on these 22 master events is then applied to all 1245 events with moment magnitude ranges between 1.7 to -0.6. We compare our predicted locations to locations computed using a diffraction stacking (DS) method (Anikiev

DOI: 10.1190/geo2020-0807.1

et al., 2014) and DS locations that were refined in a second step using a relative location (RL) method (Grechka et al., 2015). The event locations obtained by DS and RL are much better constrained in depth, i.e. around the injection depth, compared to the DS locations (Alexandrov et al., 2020).

Our results show a good localization accuracy for the events with moment magnitudes above 0.2 and potential for localizing weaker Mw events, therefore suggesting that: (1) synthetic data can be used to train CNNs to localize events given field data, (2) synthetic data can be used to augment field data sets, and (3) the 3D Gaussian distribution output could be used to restrict the search space for grid-search-based source-localization methods. Additionally, since after training an output is generated within milliseconds, we believe this to be an initial step towards real-time source localization either as a post-processing step or during the actual operations in situations where these last over several weeks.

The paper is structured as follows. First, we describe how our deep learning (DL) model is trained and discuss some of the basic operations performed in convolutional neural networks. Here we also introduce the modified U-Net architecture that we use to reconstruct the source locations in terms of 3D Gaussian distributions. In the next section we introduce the field as well as the synthetic data used to train the network. This is followed by the localization results obtained using different types of learning strategies as well as data augmentations. Additionally, we show how the locations are affected by the value of Mw. Finally, we discuss the results of previous analyses and summarize our results.

METHODS

In this section we introduce U-Nets that are based on convolutional neural networks (CNNs). This is followed by the U-Net architecture used for this study. Finally, we briefly describe the training process.

U-Nets

U-Nets have a down-sampling path (encoder) and an up-sampling path (decoder) where the number of layers in both these paths are approximately the same. Furthermore, U-Nets contain skip connections between each encoder and decoder layer, which are used to pass information from the encoder to the decoder.

The convolutional layers take the output from the previous layer as input and compute convolutions of the input with a set of filters. The coefficients of these filters are optimized during the training stage. The input can be down-sampled by writing the crosscorrelation operation with an additional parameter that determines by how many samples the filter, k, skips over the input, y, to produce the output, x. This parameter is known as stride and the operation is called strided convolution,

$$x_{i,j} = \sum_{n=1}^{K_H} \sum_{m=1}^{K_W} k_{n,m} y_{is_h+n-1, js_w+m-1}, \tag{1}$$

where s_h and s_w define the stride along the height and width of the input and K_H and K_W denote the height and width of the filter, respectively.

In the decoder the down-sampled input is gradually up-sampled using transposed convolutions. The filters used in the transposed convolutions are also optimized in

the training phase.

Training

As mentioned in the introduction, the network returns a 3D Gaussian distribution whose peak location is defined at the source location. To optimize the weights we used the sigmoid cross-entropy loss function, similar to Zhang et al. (2020b). The sigmoid cross-entropy loss, $\mathfrak{L}$ is defined as,

$$\mathfrak{L} = -\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[t_j^{(i)}\log(\hat{t}_j^{(i)}) + (1-t_j^{(i)})\log(1-\hat{t}_j^{(i)})\right], \tag{2}$$

where t_j is the target value i.e. the true 3D Gaussian distribution for the input j and $\hat{t}$ is the predicted output generated by the network. We take the sum of the sigmoid cross-entropy loss over the number of training examples, N, and the number of voxels, M. Due to the large size of the training set a stochastic gradient-descent algorithm is used. With stochastic gradient-descent, in each epoch (iteration) the entire training set is processed in smaller subsets, called a *mini-batch*. In each training step, a mini-batch of N training examples is used to compute the loss and update the weights of the network. At the beginning of every epoch the entire training set is randomly shuffled.

We used the ADAM algorithm (Kingma and Ba, 2015) to optimize the parameters in the network and the entire optimization process was implemented in Tensorflow (Abadi et al., 2015). We used the default values as proposed by Kingma and Ba

(2015), but adjusted the learning rate (or step size) to 0.001. Finally, N=20 was used as batch size.

U-Net architecture

The U-Net used in this work receives 3D tensors of seismic waveforms as input, the exact input shape being $(1024 \times 96 \times 1)$ referring to the time, seismic trace and components, respectively, the number of components being 1 here since we only have the vertical component. For each convolutional layer a specific number of filters are used, which determine the number of feature maps in that layer. In these convolutional layers the 3D tensors are described by height, width and feature maps as (height $\times$ width $\times$ feature maps). We used 32 filters in the first five convolutional layers and 64 filters in the remaining convolutional and transposed-convolutional layers. The height and width (K_H and K_W) of all filters was set to 3. In the encoder, we gradually shrink the height and width of the previous input by computing strided convolutions. With respect to our seismic input, height corresponds to time and width to seismic traces. The shapes, i.e. the height, width and number of feature maps in each layer of the network are shown in Figure 1. In the first layer following the seismic input the number of feature maps is 32 whereas the height and width stay the same as in the input. In the next layer this input of shape $(1024 \times 96 \times 32)$ is down-sampled using strided convolutions with strides of $(s_h = 2, s_w = 1)$ to a shape of $(512 \times 96 \times 32)$. We can see that the original input is down-sampled to a final size of $(8 \times 6 \times 64)$ before being up-sampled in the decoder to the final output size of $(128 \times 96 \times 64)$ representing a

3D Gaussian distribution.

As just described, the input and output of all convolutional layers are 3D tensors. We represent the output of the previous convolutional layer, $(l-1)$, as $\mathbf{z}^{(l-1)} \in \mathbb{R}^{Z_H \times Z_W \times C_i}$ with height Z_H, width Z_W, and feature maps C_i, where the subscript i stands for input. Similarly, we let the output layer, l, be $\mathbf{z}^l \in \mathbb{R}^{Z_H \times Z_W \times C_o}$ with feature maps C_o, where the subscript o stands for output. Finally, the filter $\mathbf{w}^l \in \mathbb{R}^{K_H, K_W, C_o, C_i}$ has four dimensions. A single activation unit in the l-th convolutional layer can be computed as,

$$z^{(l)}_{h,w,o} = \phi\left(\sum_{i=1}^{C_i}\sum_{n=1}^{K_H}\sum_{m=1}^{K_W}\left\{w^{(l)}_{n,m,i,o} z^{(l-1)}_{hs_h+n-1, ws_w+m-1, i}\right\}\right). \quad (3)$$

Equation 3 computes the sum of the strided convolution of the filter, $\mathbf{W}^{(l)}_{i,o}$, and the feature map of the previous layer, $\mathbf{z}^{(l-1)}_i$, and applies a nonlinear activation function $\phi(\cdot)$.

We used the (non-linear) rectified linear unit, ReLU, (Nair and Hinton, 2010) as the activation function in both the encoder and decoder, $\phi(x) = max(x, 0)$. We then apply batch normalization (Ioffe and Szegedy, 2015), which normalizes the output feature maps over the training batch by subtracting the mean and dividing by the standard deviation of the batch. In contrary to the classical U-Net architecture, which contains skip connections between each encoder and decoder layer, we only added skip connections in the deeper layers of the U-Net as represented by the horizontal arrows in Figure 1. The skip connections concatenate the feature maps from a layer of the encoder to the feature maps in one of the decoder layers (Ronneberger et al., 2015) and speed up convergence during training (Li et al., 2017). In the last layer we used

the sigmoid activation function, $S(x) = e^x/(e^x + 1)$, to map every voxel of the input, x, into a range of values between 0 and 1.

[Figure 1 about here.]

FIELD DATA AND TRAINING DATA SET

In this section we introduce the field data used to evaluate the networks that were trained with the synthetics. The field data are accompanied by an earthquake catalog and a P-wave velocity model.

Field data

The data set contains microseismic events caused by hydraulic fracturing operations from 2010 in the Barnett Shale Formation in the Fort Worth Basin in Texas, USA (Kratz et al., 2012; Alexandrov et al., 2020). The data were acquired by 543 vertical component geophones that were placed in shallow boreholes spanning an area of approximately 144 square kilometers. Each borehole was equipped with 3 geophones at 30, 45 and 60 meters below the surface. To limit the amount of data (in order to reduce memory and computational costs), we only kept the deepest geophones, which amounts to 181 geophones. From these, 85 receivers at large offsets had strongly attenuated signal and poor S/N. They have also been removed. This left us with 96 receivers covering the source region (Figure 2).

[Figure 2 about here.]

The field data set consists of 1245 detected events saved in fixed time windows of roughly 2.8 seconds each. The events were previously detected and located by Alexandrov et al. (2020) using a migration-type diffraction-stacking (DS) technique (Anikiev et al., 2014) and some locations were further refined in a second step using a relative location (RL) method (Grechka et al., 2015). We will refer to this method as DSRL. The relative locations were computed with respect to 27 master events (Alexandrov et al., 2020). 22 of those 27 master events are present in our data set and those have moment magnitudes between 0.3 and 1.6. We use those 22 master events as a validation set to chose the best performing U-Net that will be used to predict the locations of all 1245 events in the field data set. Before the data enter the network we apply a band-pass filter of 5-50 Hz and normalize the data by dividing them by their maximum amplitude value. Three examples of the field data after band-pass filtering are shown in Figure 3.

[Figure 3 about here.]

Generating synthetic data

We used the reflectivity method (Kennett and Kerry, 1979) to generate the synthetics using the open source software ERZSOL3 (Kennett, 2005), because it is fast and accurate for the situation of modeling 3D data given a 1D velocity model. The reflectivity method computes the response of layered media from a point source,

represented by the moment tensor. We defined the source region of interest within the 3D space shown in Figure 2 by the blue-shaded cuboid (it ranges from 5700 to 8300 meters in Easting, from 3700 to 5900 meters in Northing and from 1200 to 3000 meters in depth). We simulated 51200 earthquakes at random locations within the region of interest. For each event we defined a random moment tensor in terms of rake, dip and strike and therefore only consider pure double-couple sources. We limited the degrees for strike, dip and rake in the intervals [0, 360], [15, 85] and $\pm$[15, 150], respectively. We checked that including a wider range of angles did not improve the quality of the results. The source center frequency was also randomly selected in the frequency range [20, 24] Hz.

Training data set

Supervised machine learning requires an input-output pair during the learning phase. As described above, the U-Net takes a 3D tensor of seismic waveforms as input. The output is represented as a 3D Gaussian distribution with the peak at the source location defined as,

$$t(x, y, z) = exp\left(-\frac{(x - x_s)^2}{2\sigma_X^2} + \frac{(y - y_s)^2}{2\sigma_Y^2} + \frac{(z - z_s)^2}{2\sigma_Z^2}\right), \quad (4)$$

where (x_s, y_s, z_s) represent the source coordinates, (x, y, z) represent all coordinates within the 3D space of interest and σ_X, σ_Y, and σ_Z represent the spread of the Gaussian distribution in each dimension. The spread of the distribution needs to be selected prior to training. There is a trade-off between the resolution and the rate of convergence during training: The smaller the spread the sparser the 3D output

will be with most values being nearly zero with a few higher values around the true source location. This will lead to poor convergence as the loss function compares the voxel-wise difference between the output and the true distribution. On the contrary a large spread decreases resolution but increases convergence. We tested different spreads and found a spread of 200 m to yield best results. The 3D output space extends from 5500 to 8500 meters in Easting, from 3500 to 6100 meters in Northing and from 1000 to 3200 meters in depth. Thus, the size of a voxel is 23 m in Easting, 27 m in Northing and 34 m in depth.

Data augmentations

To simulate more realistic data, we applied a few data augmentations to the synthetic data during training. In addition to fixed time windows containing detected events, we have an additional 4 hours of continuous data, from which we selected noise windows where no events were detected according to the catalog. This selected field noise can be used to augment the training set during training. Since undetected events could be present within the selected field noise windows we randomly perturb the noise windows by flipping, permutating and time-shifting the individual field-noise traces.

The following data augmentations were applied to the synthetic input data: First, we randomly bulk time-shift the data. Second, we added random band-pass filtered Gaussian noise of varying intensity in each trace. Gaussian noise does not represent field noise and is of limited use if strong field noise is present in the data. However, it is a good way to avoid zero-valued entries before and after the event in the synthetics

DOI:10.1190/geo2020-0863.1

and to artificially augment the size of the data set. Third, we add continuous field noise to the data. As a final augmentation we applied station-dropout as proposed by Kriegerowski et al. (2019), which means that we randomly mute between 5 and 20 traces. A few training examples with applied data augmentations are shown in Figure 4. After applying all data augmentations the data is normalized by its maximum absolute value before it is passed to the network.

[Figure 4 about here.]

Training and validation set and evaluation metrics

We use the 51200 synthetic waveforms and their known locations as a training set and the 22 master events as validation set. The purpose of the validation set is to ensure that the network is not overfitting to the training data and to select the best performing U-Net. To this end we evaluate the performance of the trained model on the validation set. If the discrepancy in performance between the two is high, with much better performance for the training set, the network is said to be overfitting. We used the Dice-similarity coefficient (Dice, 1945) (DSC) as evaluation metric,

$$\mathrm{DSC}(t, \hat{t}) = 2\frac{t \cap \hat{t}}{t + \hat{t}}, \tag{5}$$

where again t is the target (true 3D Gaussian distribution), $\hat{t}$ is the 3D output distribution predicted by the model and $\cap$ is the symbol for intersection. Before computing the DSC we clip all values in the target and network output distribution with values above a threshold of 0.1 to 1 and the rest to 0.

DOI: 10.1190/geo2020-0807.1

RESULTS

In this section, we present the results of the trained U-Nets applied to the field data. We start by showing the results for the U-Net described earlier, which was trained on synthetic data using all data augmentation steps described before, applied to the 22 master events selected as validation set. We will refer to this model, trained with all data augmentation steps (including field noise) and with skip connections, as U-Net A. Next, we show and discuss the results for (1) a U-Net trained without field noise, U-Net B and (2) a model trained without skip connections, i.e., an FCN. U-Net A is our baseline model, to which we compare the results of the other models.

Since the optimization method is stochastic and therefore gives slightly different results each time, we trained each deep neural network (DNN) ten times for 20 epochs. We evaluate the DNNs using the validation set by computing the DSC-coefficient between the predicted output and the desired 3D Gaussian distribution, which is constructed using the location given by the diffraction-stacking method.

U-Net A: Baseline model

We apply the synthetic-data trained model to the 22 master events and compute the DSC value between the predicted and desired output, using the master locations computed by diffraction stacking to construct the desired output. The DSC-value over those 22 master events is 0.78, which is close to the DSC-value reached at the last epoch of training, being 0.81. To visualize the results, we take cross-sections along the

horizontal and vertical planes of the 3D output at the maximum value of the Gaussian distribution. The predicted Gaussian distributions are shown in Figure 5. The black dots denote the DS-locations and the white stars mark the peaks of the predicted Gaussian distributions. We can observe that all 22 events are well localized within the Gaussian distribution. Larger differences are observed in depth compared to the epicenter locations, which could be due to the surface acquisition and/or differences between the synthetic and real waveforms. We estimate the hypocenter from the peak of the Gaussian distribution and compare with the hypocenter computed by DS. The results of the mean peak value of the predicted output and the mean hypocenter, epicenter and focal depth differences are summarized in Table 1. The mean depth, epicenter and hypocenter differences using the baseline model are 96, 82 and 135 meters, respectively.

[Figure 5 about here.]

U-Net B: no field noise

U-Net B is trained with the same U-Net architecture as U-Net A, however without the addition of field noise in the data augmentation steps. The strength of the added gaussian noise was augmented by a factor of 4 for U-Net B compared to the other U-Nets since otherwise the noise level would be too low and, therefore, the model would do poorly on most of the master events. The DSC-value on the training set is 0.90 whereas on the 22 master events it is only 0.48. The high DSC-value observed

over the training set comes from the fact that the data are less complex (no added field noise) and therefore the learning process is simpler. This also explains the low DSC-value on the master events, which is due to the training set not representing the field data set well enough. The mean peak value over the master events is also low and the hypocenter and focal depth distances are significantly higher compared to U-Net A, see Table 1.

[Table 1 about here.]

FCN: no skip connections

We trained the FCN with the same architecture as the U-Net and with all data augmentation steps but without skip connections. The DSC-value at the end of training on the training set is 0.83 whereas on the master events it is quite a bit lower, i.e. 0.70. The mean depth, epicenter and hypocenter differences are 177, 54 and 163 meters, respectively. The epicenter locations are closer to the catalog locations compared to all other models, however, the mean depth as well as hypocenter differences are larger compared to U-Net A.

U-Net A applied to all detected field data events

From the above results we see that the peak value can change depending on the input data. For the model trained without field noise some of the predictions had peak values around 0.1 and the predicted output did not resemble a 3D Gaussian

distribution. Thus the peak value could be used as an indicator of how reliably the model recognizes waveforms in the input data. Since we train the deep learning algorithm on synthetic data we do expect more difficulties for the models to recognize and therefore accurately predict locations of lower magnitude events with lower S/Ns. A threshold acting on the peak value could be set to only consider predictions passing the threshold. We apply U-Net A to all 1245 field data events and consider the location at the peak value of the output as the predicted hypocenter location and compute its distance from the hypocenter location given in the catalog that is based on DS and further refined using the master events with the RL method. In Fig. 6 the peak value with respect to magnitude of all 1245 events is plotted. We observe a trend between the peak value returned by U-Net A and the Mw. Each data point is colored according to its distance to the cataloged location (based on DSRL). Most events predicted at distances smaller than 200 meters show higher peak values and also have higher magnitudes. A majority of the events with large distances between the catalog and predicted locations have peak values below 0.3 with moment magnitudes between -0.6 to 0.3. We thus decide to set the threshold at 0.4.

[Figure 6 about here.]

We compare the predicted locations passing the threshold to the locations determined by the diffraction stacking method (DS) (Anikiev et al., 2014), catalog DS, and to the locations determined by the diffraction stacking and relative-location method (RL), catalog DSRL. The main difference between the locations in catalog DS and catalog DSRL are the depth locations, which in the latter are much more concen-

trated along the injection depth level. For a detailed comparison between the two catalogs applied to the Texas data set we refer to (Alexandrov et al., 2020). To view the differences in locations between the catalogs and the predicted locations we draw lines connecting the predicted to both cataloged locations of each event (see Fig. 7). While some predicted locations still show larger differences to the cataloged locations most have a close match. A grid pattern is observed in the locations returned by U-Net A, which is due to the discretized 3D output space. We note that depth locations given by DSRL are concentrated around 2100 m depth, whereas the focal depths given by DS as well as U-Net A are scattered around 2100 m depth. Comparing the depth distribution of the located events in a histogram (Fig. 8) reveals that the depth distribution predicted by U-Net A more closely follows the trend of the DSRL catalog compared to the DS catalog, with most events at depths between 2100 and 2200 meters, which is the injection depth level. This could indicate that U-Net A is doing better at predicting the depth locations compared to DS for the events passing the threshold. The mean hypocenter, epicenter and depth distances of the locations predicted by U-Net A compared to the DSRL catalog are 214, 147 and 130 m, respectively for the 467 events that passed the threshold. The moment magnitude of those events range between -0.4 and 1.7.

[Figure 7 about here.]

[Figure 8 about here.]

With the threshold of 0.4 there are still some events that have large location differences compared to the catalogs. To focus on the predicted locations that more closely match the DSRL cataloged locations we set the threshold to 0.6 (as can be seen in Fig. 6). Comparing these predicted event locations to both cataloged locations, we observe a good match in epicenter locations (Fig. 9). Furthermore, the predicted depth locations are more concentrated around the expected depth level compared to DS. A total of 314 events pass that threshold and the mean hypocenter, epicenter and depth distances compared to the DSRL catalog are 163, 110 and 99 m, respectively, with moment magnitudes in the range -0.2 to 1.7.

Since the locations from the catalog are captured inside the 3D Gaussian distribution returned by the U-Net, the U-Net's locations could be used as initial source locations that can then be further improved with the use of other microseismic source localization methods such as RL.

[Figure 9 about here.]

DISCUSSION

We did not address the issue of the *detection* of events in this study since we do not have continuous field data. For a practical application we would suggest to separate the detection and localization problem. The detection could for instance be made by diffraction stacking (Anikiev et al., 2014; Staněk et al., 2015) or other stacking methods (Loginov et al., 2016). Machine learning methods capable of differentiating

DOI: 10.1190/geo2020-0863.1

between signal and noise have also already been proposed and successfully applied (Perol et al., 2018; Mousavi et al., 2019; Li et al., 2018; Wu et al., 2018) and could also be used in combination with an event-localization method. If an event is detected, the information can be passed to the source-localization network. Alternatively, we could explore whether the U-Net trained in this study could be used as a detection method, based on the peak value returned by the model. In continuous mode a possible criterion for a detection could be that the peak value should pass a predefined threshold in a number of consecutive time windows if an event is present.

One main limitation of the method is the ability to mimic realistic field noise that can be added to the synthetics - especially to target low-S/N events. Each station in the field is subjected to local noise. Thus, if longer passive noise recordings were available for each station, these could easily be added to the corresponding traces in the synthetics during training. The network would then be able to learn directly with the noise specific to each station and possibly be able to detect and predict the hypocenter locations of lower S/N-events. This subject falls outside the scope of this work.

Contrary to previous works on earthquake source localizations using CNNs (Kriegerowski et al., 2019; Zhang et al., 2020b; Mousavi and Beroza, 2020b; van den Ende and Ampuero, 2020) this work only made use of the vertical component of the seismic wavefield, because this was the only component recorded in the field. It would be interesting to study the differences of a network trained with both the vertical and horizontal wavefields and a network trained only with the vertical component. We

can assume that the network with all three components would perform better since many source-localization methods make use of both P- and S-waves.

In this study, a subset of all available stations was used to reduce the amount of data needed to store the synthetic data and reduce the data input-output bottlenecks and memory footprint during the training phase of the network. However, the earthquake catalog with the source locations is based on all the available stations. Even though less stations were used the network still returned accurate source locations.

In this study we did not address the problem of localizing multiple events present in a single time window and only considered the situation where a single event is present, as this was the case in the field data. To address this issue we would suggest training a network with input data containing a random number of events, K, and similarly the target would therefore contain K Gaussian distributions. In that way the network could possibly learn to recognize whether more than a single event is present in the input and return source locations for those events.

Generating the synthetic data set for 51200 sources took 7 days running 100 jobs in parallel on 2.3 GHz Intel Xeon CPUs. Training for 20 epochs took 11 hours on a NVIDIA GeForce GTX 1080 Ti GPU. Prediction of a single event takes 0.28 seconds on a 3.1 GHz Dual-Core Intel Core i5 processor.

In this work we extended upon previous works using CNNs (Perol et al., 2018; Kriegerowski et al., 2019; Zhang et al., 2020b) and specifically addressed the problem of missing events in the training set by generating synthetics everywhere in the model space. We wanted to test the possibility of training a network solely with

synthetic data that can directly be applied to field data. This work shows that this is indeed possible, however, the ability of localizing weaker magnitude field data events decreases rapidly. It could be investigated whether augmenting the synthetics using longer recordings of passive noise from the area under investigation could lead to better localizations of weaker events or whether the weights of the network trained with the synthetic data set can be fine-tuned using field data in order to localize weaker magnitude events.

In this study we applied the method to single-component data from a hydraulic fracturing site, however, it can be applied to larger areas by up-scaling the entire experimental setup.

CONCLUSION

In this paper we showed that synthetic data can be used to train a U-Net to accurately localize microseismic field data. Furthermore, we showed that augmenting the synthetic data with field noise further increases the U-Net's accuracy to localize events. After the network is trained, this method returns the source location within less than one second given the event waveforms as input. Furthermore, the retrieved locations are comparable to state-of-the-art localization methods such as diffraction-stacking and refined diffraction-stacking locations using a relative location method. In terms of depth locations the deep learning model seems to outperform diffraction-stacking as the depths are better constrained around the expected depth level for the events predicted with peak values above a set threshold. The proposed method

provides locations based on the full waveform without the need for any picking, while its accuracy was better than that of a conventional diffraction stacking approach.

ACKNOWLEDGMENTS

We would like to thank Dmitry Alexandrov and Leo Eisner from the company *Seismik s.r.o.* for providing us with the Texas data set, their earthquake catalog and the velocity model.

REFERENCES

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (Software available from tensorflow.org).

Alexandrov, D., L. Eisner, U. b. Waheed, S. I. E. Kaka, and S. A. Greenhalgh, 2020, Normal faulting activated by hydraulic fracturing: A case study from the Barnett Shale, Fort Worth Basin: The Leading Edge, **39**, 204–211.

Anikiev, D., J. Valenta, F. Staněk, and L. Eisner, 2014, Joint location and source mechanism inversion of microseismic events: Benchmarking on seismicity induced by hydraulic fracturing: Geophysical Journal International, **198**, 249–258.

Bishop, C. M., 1994, Mixture density networks.

———, 1995, Neural networks for pattern recognitions: Oxford University Press.

Chambers, K., B. D. Dando, G. A. Jones, R. Velasco, and S. A. Wilson, 2014, Moment tensor migration imaging: Geophysical Prospecting, **62**, 879–896.

Chen, H., X. Meng, F. Niu, Y. Tang, C. Yin, and F. Wu, 2018, Microseismic monitoring of stimulating shale gas reservoir in SW China: 2. Spatial clustering controlled by the preexisting faults and fractures: Journal of Geophysical Research: Solid

Earth, **123**, 1659–1672.

Dice, L. R., 1945, Measures of the amount of ecologic association between species: Ecology, **26**, 297–302.

Grechka, V., A. De La Pena, E. Schisselé-Rebel, E. Auger, and P.-F. Roux, 2015, Relative location of microseismicity: Geophysics, **80**, WC1–WC9.

Huang, L., J. Li, H. Hao, and X. Li, 2018, Micro-seismic event detection and location in underground mines by using Convolutional Neural Networks (CNN) and deep learning: Tunnelling and Underground Space Technology, **81**, 265–276.

Ioffe, S., and C. Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift: arXiv preprint arXiv:1502.03167.

Johnson, S. W., D. J. Chambers, M. S. Boltz, and K. D. Koper, 2021, Application of a convolutional neural network for seismic phase picking of mining-induced seismicity: Geophysical Journal International, **224**, 230–240.

Jozinović, D., A. Lomax, I. Štajduhar, and A. Michelini, 2020, Rapid prediction of earthquake ground shaking intensity using raw waveform data and a convolutional neural network: Geophysical Journal International, **222**, 1379–1389.

Kao, H., R. Visser, B. Smith, and S. Venables, 2018, Performance assessment of the induced seismicity traffic light protocol for northeastern British Columbia and western Alberta: The Leading Edge, **37**, 117–126.

Käufl, P., A. P. Valentine, R. W. de Wit, and J. Trampert, 2016a, Solving probabilistic inverse problems rapidly with prior samples: Geophysical Journal International, **205**, 1710–1728.

Käufl, P., A. P. Valentine, and J. Trampert, 2016b, Probabilistic point source inver-

DOI: 10.1190/geo2020-0685.1

sion of strong-motion data in 3-D media using pattern recognition: A case study for the 2008 Mw 5.4 Chino Hills earthquake: Geophysical Research Letters, **43**, 8492–8498.

Kennett, B. L., 2005, ERZSOL3 (Reflectivity Method): `http://www.quest-itn.org/library/software/reflectivity-method.html`. (Accessed: 2020-03-30).

Kennett, B. L., and N. J. Kerry, 1979, Seismic waves in a stratified half space: Geophysical Journal of the Royal Astronomical Society, **57**, 557–583.

Kingma, D. P., and J. L. Ba, 2015, Adam: A method for stochastic optimization: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 1–15.

Kratz, M., A. Aulia, and A. Hill, 2012, Identifying fault activation in shale reservoirs using microseismic monitoring during hydraulic stimulation: Source Mechanisms, b Values, and energy release rates: CSEG Recorder, **20**.

Kriegerowski, M., G. M. Petersen, H. Vasyura-Bathke, and M. Ohrnberger, 2019, A deep convolutional neural network for localization of clustered earthquakes based on multistation full waveforms: Seismological Research Letters, **90**, 510–516.

Li, H., Z. Xu, G. Taylor, C. Studer, and T. Goldstein, 2017, Visualizing the loss landscape of neural nets: arXiv preprint arXiv:1712.09913.

Li, L., J. Tan, B. Schwarz, F. Staněk, N. Poiata, P. Shi, L. Diekmann, L. Eisner, and D. Gajewski, 2020, Recent advances and challenges of waveform-based seismic location methods at multiple scales: Reviews of Geophysics, **58**.

Li, L., J. Tan, D. A. Wood, Z. Zhao, D. Becker, Q. Lyu, B. Shu, and H. Chen, 2019, A review of the current status of induced seismicity monitoring for hydraulic

fracturing in unconventional tight oil and gas reservoirs: Fuel, **242**, 195–210.

Li, Z., M.-A. Meier, E. Hauksson, Z. Zhan, and J. Andrews, 2018, Machine learning seismic wave discrimination: Application to earthquake early warning: Geophysical Research Letters, **45**, 4773–4779.

Loginov, G., A. Duchkov, and F. Andersson, 2016, The structure-tensor analysis for optimal microseismic data partial stack: 78th EAGE Conference and Exhibition 2016, European Association of Geoscientists & Engineers, 1–5.

Lomax, A., A. Michelini, and D. Jozinović, 2019, An investigation of rapid earthquake characterization using single-station waveforms and a convolutional neural network: Seismological Research Letters, **90**, 517–529.

Long, J., E. Shelhamer, and T. Darrell, 2015, Fully convolutional networks for semantic segmentation: Proceedings of the IEEE conference on computer vision and pattern recognition, 3431–3440.

Maxwell, S. C., T. Urbancic, N. Steinsberger, R. Zinno, et al., 2002, Microseismic imaging of hydraulic fracture complexity in the Barnett shale: Presented at the SPE annual technical conference and exhibition, Society of Petroleum Engineers.

McKean, S., J. Priest, J. Dettmer, and D. Eaton, 2019, Quantifying fracture networks inferred from microseismic point clouds by a Gaussian mixture model with physical constraints: Geophysical Research Letters, **46**, 11008–11017.

Mendecki, A. J., 1993, Keynote address: Real time quantitative seismology in mines: Rockbursts and Seismicity in Mines, 287–295.

Mousavi, S. M., and G. C. Beroza, 2020a, A machine-learning approach for earthquake magnitude estimation: Geophysical Research Letters, **47**, e2019GL085976.

———, 2020b, Bayesian-Deep-Learning Estimation of Earthquake Location From Single-Station Observations: IEEE Transactions on Geoscience and Remote Sensing, **58**, 8211–8224.

Mousavi, S. M., S. P. Horton, C. A. Langston, and B. Samei, 2016, Seismic features and automatic discrimination of deep and shallow induced-microearthquakes using neural network and logistic regression: Geophysical Journal International, **207**, 29–46.

Mousavi, S. M., W. Zhu, Y. Sheng, and G. C. Beroza, 2019, CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection: Scientific reports, **9**, 1–14.

Münchmeyer, J., D. Bindi, U. Leser, and F. Tilmann, 2021, The transformer earthquake alerting model: a new versatile approach to earthquake early warning: Geophysical Journal International, **225**, 646–656.

Nair, V., and G. E. Hinton, 2010, Rectified linear units improve restricted Boltzmann machines: Proceedings of the 27th international conference on machine learning (ICML-10), 807–814.

Pearson, C., 1981, The relationship between microseismicity and high pore pressures during hydraulic stimulation experiments in low permeability granitic rocks: Journal of Geophysical Research: Solid Earth, **86**, 7855–7864.

Perol, T., M. Gharbi, and M. Denolle, 2018, Convolutional neural network for earthquake detection and location: Science Advances, **4**, 2–10.

Rodríguez-Pradilla, G., and D. W. Eaton, 2020, Automated Microseismic Processing and Integrated Interpretation of Induced Seismicity during a Multistage Hydraulic-

Fracturing Stimulation, Alberta, Canada: Bulletin of the Seismological Society of America, **110**, 2018–2030.

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, Springer, 234–241.

Ross, Z. E., M.-A. Meier, and E. Hauksson, 2018a, P wave arrival picking and first-motion polarity determination with deep learning: Journal of Geophysical Research: Solid Earth, **123**, 5120–5129.

Ross, Z. E., M.-A. Meier, E. Hauksson, and T. H. Heaton, 2018b, Generalized seismic phase detection with deep learning: Bulletin of the Seismological Society of America, **108**, 2894–2901.

Simpson, D. W., W. Leith, and C. Scholz, 1988, Two types of reservoir-induced seismicity: Bulletin of the Seismological Society of America, **78**, 2025–2040.

Staněk, F., D. Anikiev, J. Valenta, and L. Eisner, 2015, Semblance for microseismic event detection: Geophysical Journal International, **201**, 1362–1369.

Trojanowski, J., and L. Eisner, 2017, Comparison of migration-based location and detection methods for microseismic events: Geophysical Prospecting, **65**, 47–63.

van den Ende, M. P., and J.-P. Ampuero, 2020, Automated seismic source characterization using deep graph neural networks: Geophysical Research Letters, **47**, e2020GL088690.

Van Der Baan, M., D. Eaton, M. Dusseault, et al., 2013, Microseismic monitoring developments in hydraulic fracture stimulation: Presented at the ISRM international conference for effective and sustainable hydraulic fracturing, International Society

for Rock Mechanics and Rock Engineering.

Wu, Y., Y. Lin, Z. Zhou, D. C. Bolton, J. Liu, and P. Johnson, 2018, DeepDetect: A cascaded region-based densely connected network for seismic event detection: IEEE Transactions on Geoscience and Remote Sensing, **57**, 62–75.

Zhang, G., C. Lin, and Y. Chen, 2020a, Convolutional neural networks for microseismic waveform classification and arrival picking: Geophysics, **85**, WA227–WA240.

Zhang, X., J. Zhang, C. Yuan, S. Liu, Z. Chen, and W. Li, 2020b, Locating induced earthquakes with a network of seismic stations in Oklahoma via a deep learning method: Scientific Reports, **10**, 1–14.

Zhu, L., Z. Peng, J. McClellan, C. Li, D. Yao, Z. Li, and L. Fang, 2019, Deep learning for seismic phase detection and picking in the aftershock zone of 2008 Mw7. 9 Wenchuan Earthquake: Physics of the Earth and Planetary Interiors, **293**, 106261.

Zhu, W., and G. C. Beroza, 2019, PhaseNet: a deep-neural-network-based seismic arrival-time picking method: Geophysical Journal International, **216**.

LIST OF FIGURES

Table 1: Mean peak value of DNNs over 22 master events and their location differences compared to DS catalog.

Model	Peak value	Hypocenter difference (m)	Epicenter difference (m)	Focal depth difference (m)
U-Net A	0.79	135	82	96
U-Net B	0.59	302	175	223
FCN	0.75	177	54	163

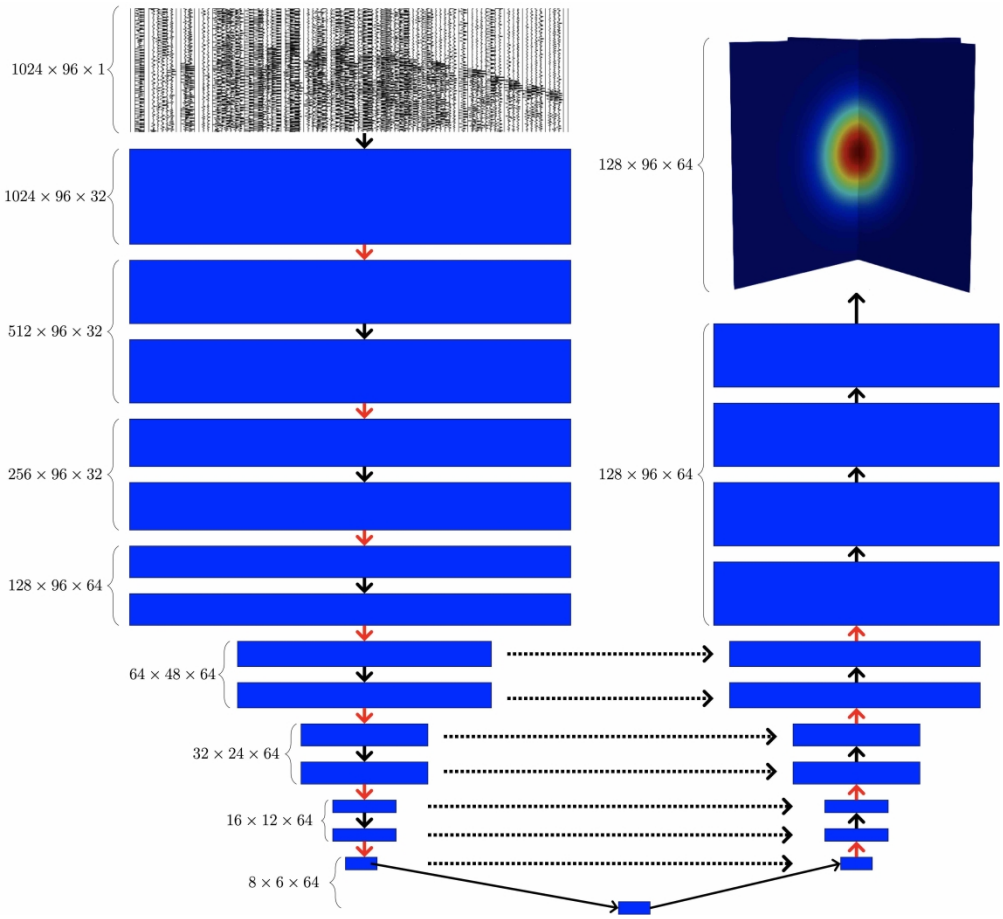

Figure 1: Illustration of the U-Net architecture used throughout this work: seismic input data go through a set of convolutional and transposed convolutional layers generating a 3D Gaussian distribution as output. Black down- and up-going arrows indicate standard convolutional layers. Red down-going arrows denote strided convolutions (down-sampling) and up-going arrows denote transposed convolutional layers (up-sampling). Dashed horizontal lines indicate skip connections between encoder and decoder. The numbers next to the curly brackets indicate the dimensions of each layer.

945x862mm (144 x 144 DPI)

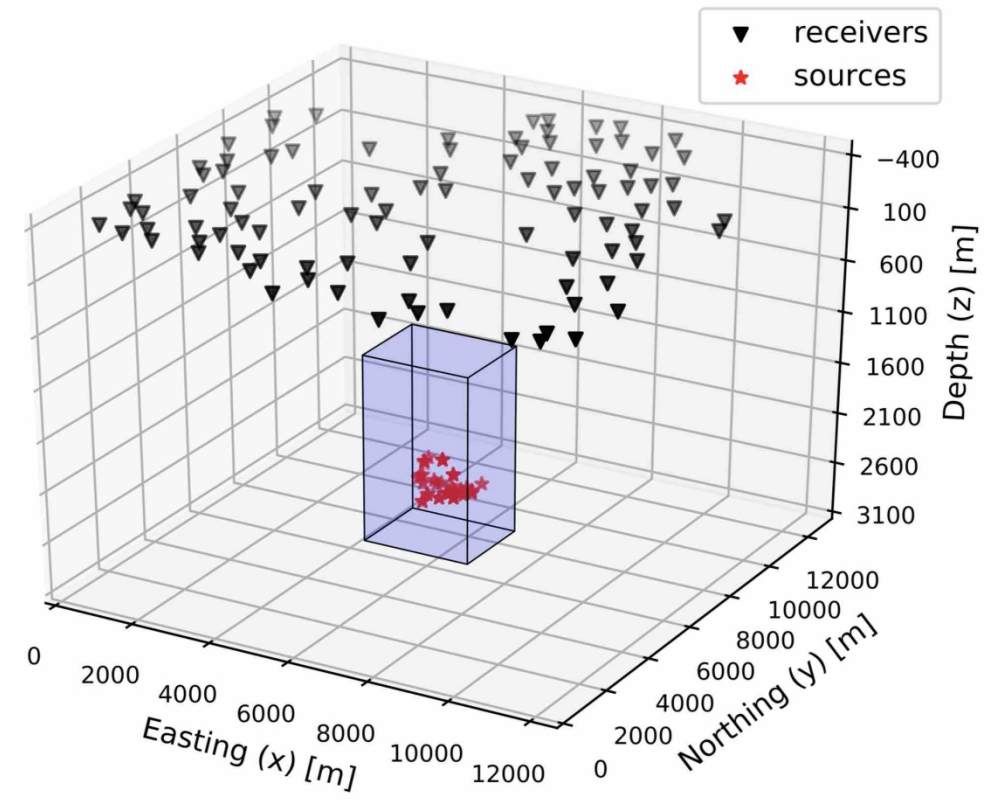

Figure 2: Receiver locations (black triangles), source region of interest (shaded cuboid) and source locations (red stars).

360x292mm (144 x 144 DPI)

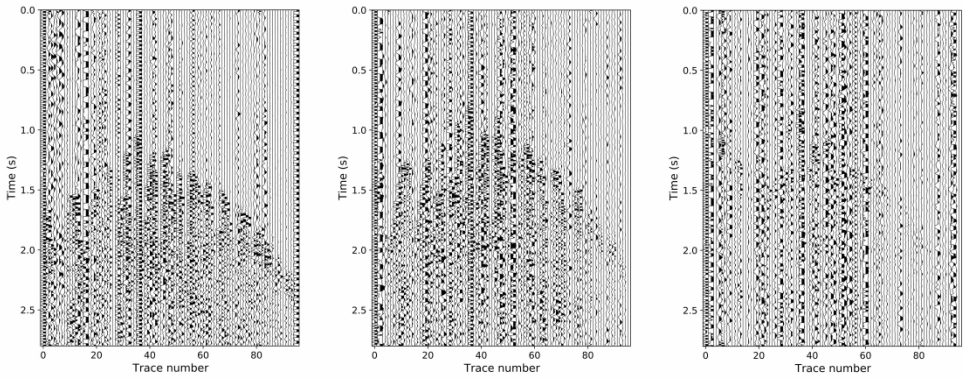

Figure 3: Bandpass-filtered examples of three master events with decreasing signal quality from left to right. Percentile clipping was applied to better visualize events. Shown entire waveforms are used as input to the U-Nets (without added percentile clipping).

1370x609mm (144 x 144 DPI)

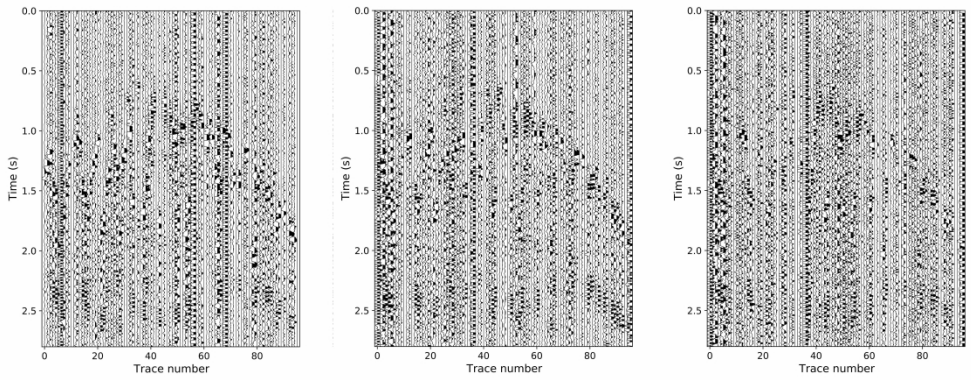

Figure 4: Three synthetic events augmented with field noise as used during training.

1371x609mm (144 x 144 DPI)

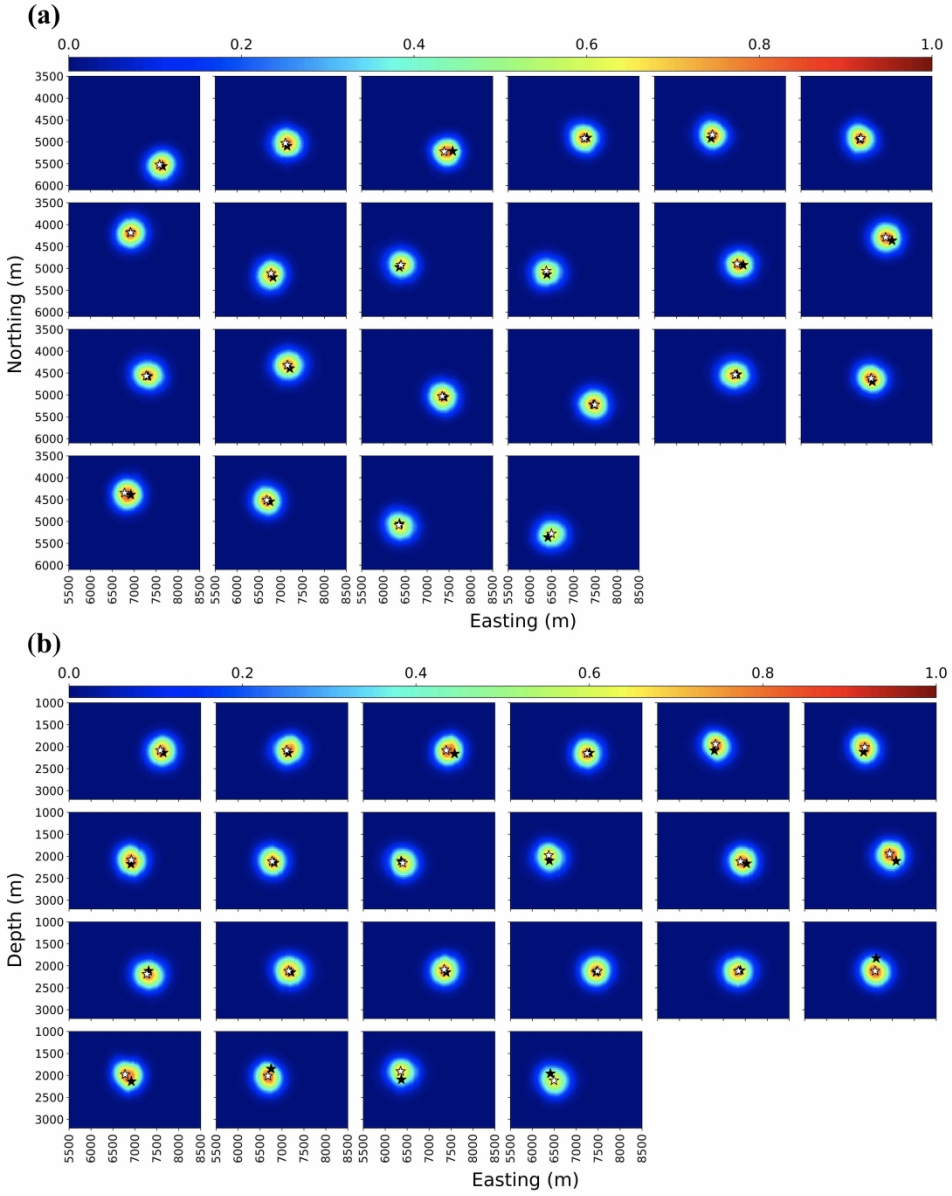

Figure 5: U-Net A output cross-sections: (a) horizontal cross-section with Easting along horizontal and Northing along vertical axis and (b) vertical cross-section with Easting along horizontal and depth along vertical axis. Black stars represent DS cataloged locations and white stars are placed at highest-valued voxel in U-Net A's output. Each panel corresponds to output for one of the 22 master events with Mw between 0.3 and 1.6.

891x1129mm (144 x 144 DPI)

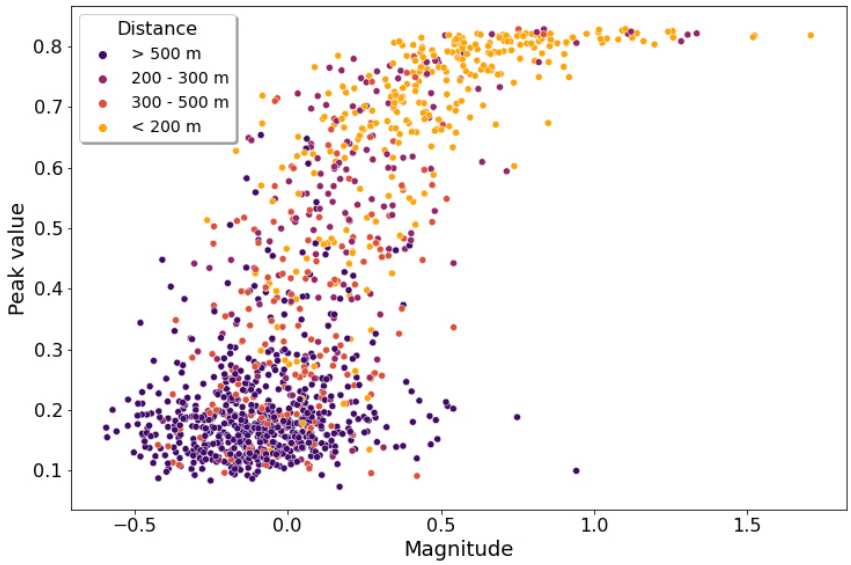

Figure 6: Peak value of U-Net A output vs Magnitude of 1245 field data events. Each data point is colored according to its distance to the DSRL cataloged location.

783x522mm (28 x 28 DPI)

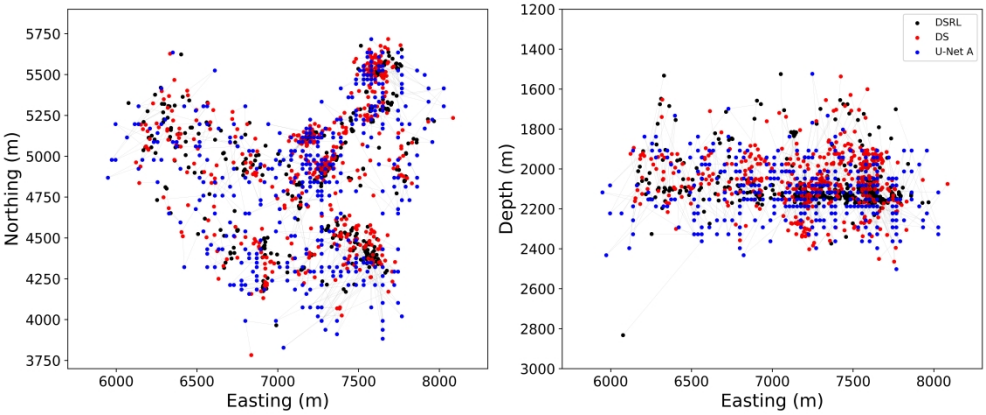

Figure 7: Locations given in DS catalog (red), DSRL catalog (black) and locations predicted by U-Net A passing an amplitude threshold of 0.4 (blue). Lines connect predicted to cataloged locations.

904x387mm (236 x 236 DPI)

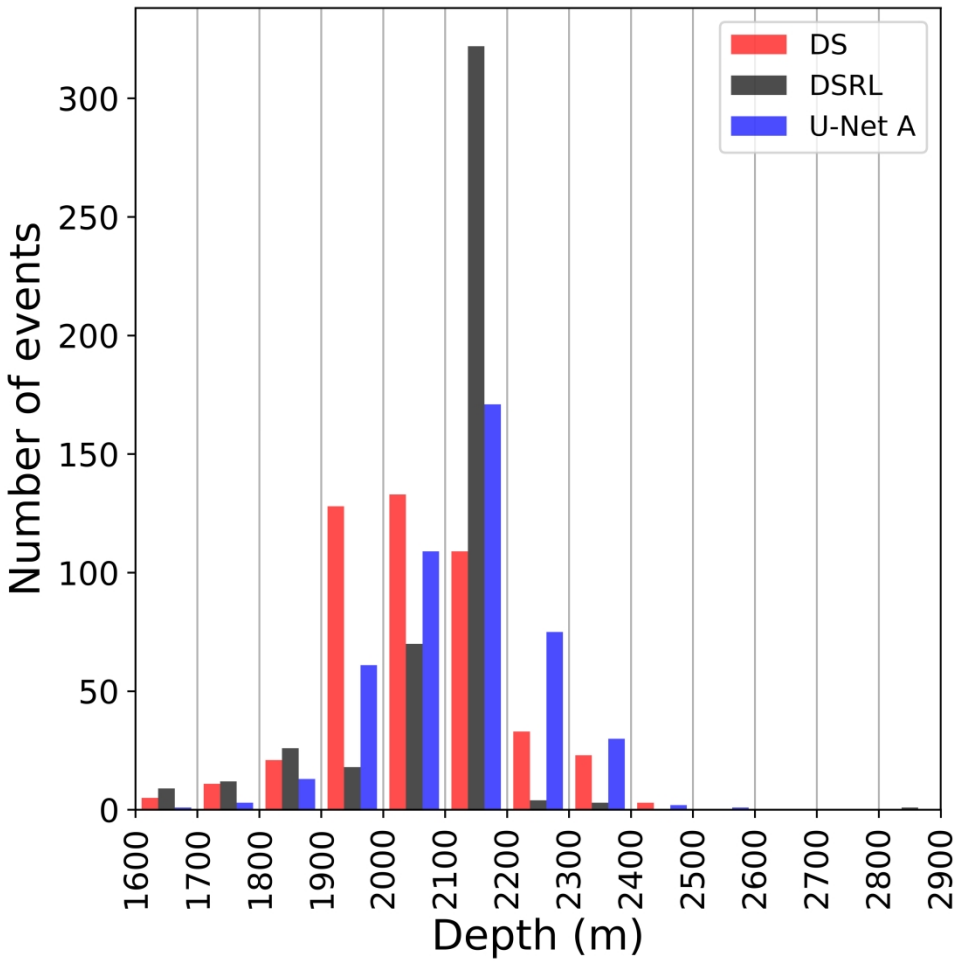

Figure 8: Depth distributions of events in DS catalog (red), DSRL catalog (black) and locations predicted by U-Net A passing an amplitude threshold of 0.4 (blue).

387x387mm (300 x 300 DPI)

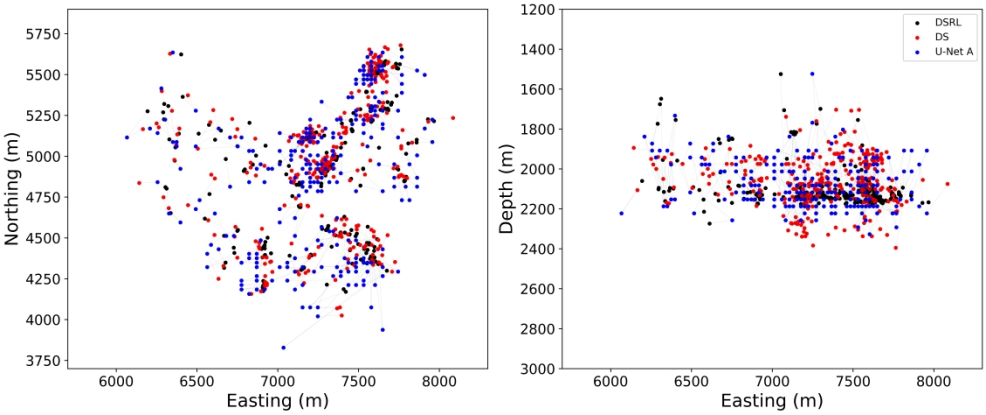

Figure 9: Locations given in DS catalog (red), DSRL catalog (black) and locations predicted by U-Net passing threshold of 0.6 (blue). Lines connect predicted to both cataloged locations.

904x387mm (236 x 236 DPI)

DATA AND MATERIALS AVAILABILITY

Custom statement of data and materials availability

DOI:10.1190/geo2020-0868.1