GEOPHYSICS, VOL. 85, NO. 5 (SEPTEMBER-OCTOBER 2020); P. KS149–KS160, 14 FIGS., 6 TABLES.
10.1190/GEO2019-0774.1

DOI:10.1190/geo2019-0774.1

Application of machine learning to microseismic event detection in distributed acoustic sensing data

Anna L. Stork[1], Alan F. Baird[2], Steve A. Horne[3], Garth Naldrett[4], Sacha Lapins[2], J.-Michael Kendall[5], James Wookey[2], James P. Verdon[2], Andy Clarke[4], and Anna Williams[2]

ABSTRACT

This study presents the first demonstration of the transferability of a convolutional neural network (CNN) trained to detect microseismic events in one fiber-optic distributed acoustic sensing (DAS) data set to other data sets. DAS increasingly is being used for microseismic monitoring in industrial settings, and the dense spatial and temporal sampling provided by these systems produces large data volumes (approximately 650 GB/day for a 2 km long cable sampling at 2000 Hz with a spatial sampling of 1 m), requiring new processing techniques for near-real-time microseismic analysis. We have trained the CNN known as YOLOv3, an object detection algorithm, to detect microseismic events using synthetically generated waveforms with real noise superimposed. The performance of the CNN network is compared to the number of events detected using filtering and amplitude threshold (short-term average/long-term average) detection techniques. In the data set from which the real noise is taken, the network is able to detect >80% of the events identified by manual inspection and 14% more than detected by standard frequency-wavenumber filtering techniques. The false detection rate is approximately 2% or one event every 20 s. In other data sets, with monitoring geometries and conditions previously unseen by the network, >50% of events identified by manual inspection are detected by the CNN.

INTRODUCTION

Distributed acoustic sensing

In recent years, seismic monitoring has seen the development of a novel sensing technology: distributed acoustic sensing (DAS). This technology makes use of Rayleigh scattering of laser light in fiber-optic cables to measure strain-rate on a fiber (see Hartog, 2017 for an introduction). A fiber interrogator emits a laser pulse down the fiber and records the back-scattered light (Figure 1). The phase-difference between the back-scattered light from two points in the fiber is analyzed for each channel (equivalent to a receiver). The distance between the two points is termed the gauge length. Changes in strain along the fiber caused by, for example, the passing of a seismic wave, result in changes in the recorded signal (Hartog, 2017; Ning and Sava, 2018). One significant advantage of DAS is that fibers can be kilometers long and channels are closely spaced, on the order of meters; therefore, thousands of measurements are obtained, providing dense sampling of the wavefield. For example, assuming a seismic velocity of 2500 m/s and a signal frequency of 100 Hz, a 2000 Hz sampling frequency and a 1 m channel spacing produce 25 samples per wavelength and 80 samples per period. Compared with traditional geophone or seismometer sensors, where tens (or maybe hundreds in a dense experiment) of measurements are made, fiber provides a more complete picture of a seismic wavefield, thereby providing significantly more information on, for example, geologic structure (e.g., Dou et al., 2017; Jousset et al.,

Manuscript received by the Editor 21 November 2019; revised manuscript received 29 April 2020; published ahead of production 13 June 2020; published online 17 August 2020.
[1]Formerly University of Bristol, School of Earth Sciences, Queens Road, Bristol BS8 1RJ, UK; presently Silixa Ltd., 230 Centennial Avenue, Elstree, Borehamwood WD6 3SN, UK. E-mail: anna.stork@silixa.com (corrresponding author).
[2]University of Bristol, School of Earth Sciences, Queens Road, Bristol BS8 1RJ, UK. E-mail: alan.baird@bristol.ac.uk; sacha.lapins@bristol.ac.uk; j.wookey@bristol.ac.uk; james.verdon@bristol.ac.uk; anna.williams@bristol.ac.uk.
[3]Formerly Chevron Energy Technology Company, 1 Westferry Circus, Canary Wharf, London E14 4HA, UK. E-mail: s_horne@yahoo.com.
[4]Silixa Ltd., 230 Centennial Avenue, Elstree, Borehamwood WD6 3SN, UK. E-mail: garth.naldrett@silixa.com; andy.clarke@silixa.com.
[5]Formerly University of Bristol, School of Earth Sciences, Queens Road, Bristol BS8 1RJ, UK; presently University of Oxford, Department of Earth Sciences, 3 S Parks Road, Oxford OX1 3AN, UK. E-mail: mike.kendall@stx.ox.ac.uk.

DOI:10.1190/geo2019-0774.1

2018), and, for microseismic applications, potentially providing significant improvements in event detection capability and location accuracy. Here, we examine the use of DAS in microseismic monitoring, focusing on event detection.

DAS for microseismic monitoring

The detailed information to be gained from DAS data is particularly useful for microseismic monitoring, where small seismic events (usually with magnitudes <0) are recorded in industrial settings. For example, the technology has been deployed to monitor induced seismicity at hydraulic-fracturing sites. The results from these surveys highlight the potential for enhanced monitoring by integrating temperature, strain, and microseismic measurements all made on fiber-optic cables (Karrenbach et al., 2019). DAS data provide only a 1C recording (the fiber is sensitive to changes in strain along the fiber, but it is insensitive to changes broadside to the cable), whereas standard geophones and seismometers can provide 3C recordings. Nevertheless, processing techniques are advancing to determine microseismic event locations (Webster et al., 2016; Verdon et al., 2020) and source mechanisms (Cole et al., 2018; Baird et al., 2020) from DAS data. The technology also is being explored for microseismic monitoring in other industrial settings, for example, at geothermal sites (Mondanos and Coleman, 2019), geologic CO_2 storage sites, and for volcano monitoring.

With thousands of sensors and a high sampling rate (>1000 Hz), large volumes of DAS data are recorded (e.g., approximately 650 GB/day for a 2 km-long cable sampling at 2000 Hz with a spatial sampling of 1 m). In industrial settings in which injection of fluids is taking place, data analysis often is required in near real time so operational decisions can be made based on monitoring data (Clarke et al., 2019). This is a significant challenge for DAS microseismic data; therefore, new methods are required to analyze data within an acceptable time period. Some comparisons of traditional microseismic event detection methods, such as short-term average/long-term average (STA/LTA) algorithms, have been published for DAS data (e.g., Binder and Chakraborty, 2019). However, these methods which rely on well-known signal-processing techniques often are computationally slow with the large data volumes produced by DAS systems.

Machine-learning techniques have progressed significantly in recent years, in speed and accuracy, and initial studies show potential for rapid analysis of DAS data (Binder and Chakraborty, 2019). The objective of the present study is to assess the accuracy of using a current CNN in microseismic event detection. We detail the successful retraining of a CNN known as YOLOv3 (Redmon and Farhadi, 2018), and we apply the network to event detection in multiple data sets. We compare the microseismic event detection capabilities of the retrained CNN to manual detection and the number of events detected using a classic signal processing filtering and amplitude threshold (STA/LTA) technique.

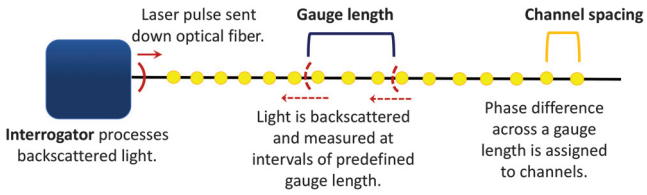

Figure 1. Schematic showing the measurement principle of DAS. A laser pulse is transmitted down the fiber. The phase difference in back-scattered light is measured over a gauge length at points separated by the channel spacing.

Machine learning

Here, we apply image recognition machine-learning techniques to DAS data to identify microseismic events. Deep CNNs are now the most popular method applied to object detection and classification (Rawat and Wang, 2017). Such methods are able to process video files in real time, and this potentially enables the real-time processing of DAS data to highlight features of interest. Benefits of using machine-learning techniques could be to reduce data storage space requirements and a reduction in data processing time for detailed analysis.

The use of machine learning in geophysics, and seismology in particular, has seen a recent increase. Earthquake detection and/or phase-picking methods mainly have concentrated on regional and global earthquake catalogs with seismic events detected on conventional networks (e.g., Ross et al., 2018; Zhu and Beroza, 2019; Zhou et al., 2019; Woollam et al., 2019) or on data from industrial settings recorded on geophones (Zhang et al., 2018). This study builds on work to detect microseismic events using Haar Cascades demonstrated by Horne et al. (2019) and, much like the study by Binder and Chakraborty (2019), this work presents the successful detection of microseismic events in DAS data using a CNN. The present study differs in the fact that we retrain an existing fast object detection algorithm suitable for use on video. Additionally, we show that the trained network is generally applicable and can be used to detect microseismic events in other data sets in which a cable is installed in a horizontal well. This is an important step in proving the feasibility of using DAS technology in microseismic monitoring and the applicability of machine learning to event detection in DAS data.

Two methods for event detection are tested in this study. The first is to train a CNN to detect microseismic events, and the second is to use filtering and amplitude threshold detection following stacking of the STA/LTA characteristic function. Each method and the results obtained using both methods are outlined below.

DATA

The data used in this study include synthetic data and also field data (Data Set 1) collected during a single hydraulic-fracturing stage. The field data are recorded using a DAS system (Silixa iDAS) and comprise 4 h of continuous data. The data are collected from a single monitoring well, parallel to the stimulation well in the horizontal section, and they are recorded using a standard fiber with 3765 channels (receivers) with 1 m spacing. The well configuration is illustrated in Figure 2. The data are recorded at 2000 Hz with a 10 m gauge length.

The synthetic microseismic event data for the study are created using the monitoring setup in Figure 2. A homogeneous vertical transverse isotropic (VTI) medium is assumed with velocities derived from cross-dipole sonic logs from the horizontal monitoring well and a nearby vertical well. The seismic velocities can be characterized by the vertical P and S velocities $V_{P0} = 2800$ m/s and $V_{S0} = 1750$ m/s and the Thomsen (1986) parameters $\epsilon = 0.42$, $\gamma = 0.36$, and $\delta = 0.21$.

DOI:10.1190/geo2019-0774.1

To produce the synthetic waveforms, a VTI ray tracer was used to compute traveltimes from a moment tensor point source to each channel for direct P, SV, and SH arrivals. Because a homogeneous model is assumed, no other arrivals were modeled. Displacement amplitudes and polarizations were computed using dynamic ray theory Green's functions derived by Chapman (2004). These Green's functions then are convolved with the time-derivative of a third-order Brune pulse source wavelet (Beresnev and Atkinson, 1997) to generate velocity synthetics. We then convert the particle velocity along the cable direction to the DAS strain rate by differencing over the gauge length and dividing by the gauge length (Miller et al., 2016). For further details on the synthetic data modeling, see Baird et al. (2020).

The source parameters for 2000 synthetic events are chosen from a random distribution of locations within a 500 m radius of the center of the microseismic cluster determined from a surface array, with moment magnitudes between −1.5 and 0.1, drawn from a Gutenberg-Richter distribution, with the source frequency dependent on the magnitude. Source mechanisms were assumed to be planar shear faulting with a randomized orientation and slip direction. The corresponding moment tensors were calculated using the equations of Vavryčuk (2005) and are largely double couple, but with some non-double couple components due to the anisotropy.

EVENT DETECTION WITH MACHINE LEARNING

Method: YOLOv3

The aim of this study is to investigate the accuracy of using a CNN to detect microseismic events in DAS data and evaluate whether a trained network could be generally applicable to event detection in microseismic monitoring data. The most efficient way to do this is to take a network trained to detect other objects and retrain it to recognize microseismic events. We use a CNN known as YOLOv3 to test the possible application because it is one of the faster object detection algorithms available and it has been reported as an accurate real-time object detector for video files (Redmon and Farhadi, 2018). Region-based object detectors (e.g., R-CNN; Ren et al., 2015) have been reported as having a small accuracy advantage over YOLOv3, but YOLOv3 is able to process a greater number of frames per second (Hui, 2018). For DAS data, microseismic event detection in (near) real-time processing is the biggest challenge, due to the large data volumes. Therefore, YOLOv3 is chosen over region-based methods. YOLOv3 is chosen over YOLOv2 because it is better able to detect smaller objects and microseismic events may be recorded on only a small portion of the fiber. Another advantage of YOLOv3 is that it is able to detect overlapping objects in an image, as often occurs in microseismic event detection.

YOLOv3 uses a single CNN. The network has been trained on the ImageNet data set to classify and locate objects in an image or video (Redmon and Farhadi, 2018). ImageNet is an online image database with hundreds of thousands of images classified by noun (the object in the image), and it often is used to test object detection algorithms (Stanford Vision Lab, 2016). For detection, a network architecture with 106 layers (75 convolutional and 31 other layers) is used (Figure A-1). Batches of images are input to the network with the number of input images being defined by the batch parameter. One batch of images is processed by the network in each iteration before the weights for the network filters are updated. Batch normalization is applied, and batches can be divided into minibatches, so GPU processes work on this number of images at once. The size of the network (width and height) is set, and input images automatically are resized to these dimensions. It is desirable that the size of the input images (and network) is large enough to retain details. For microseismic events, this detail includes identification of first-arrival times and observations of P- and S-waves and coda.

An input image is divided into a grid, and a given number of bounding boxes are predicted for each grid cell (Figure 3). A confidence score is attributed to each bounding box that reflects how likely it is that the box contains the object and how accurately the box describes the object. Using sigmoid activation, YOLOv3 predicts the probability of each class of object in each grid square. Here, we are trying to predict only one object class (a microseismic event) so each grid cell has one probability associated with it. After each iteration through the network, the loss value is given, thereby providing a measure of how well the model is predicting the existence of an object in an image. The smaller the loss, the better the model predictions are. The loss function is the function used to evaluate a candidate solution. The YOLOv3 loss function is a combination of the errors in the bounding box position (a squared error loss) and errors in the prediction of an object class (a binary cross-entropy loss) (Redmon and Farhadi, 2018; AlexeyAB, 2018). The confidence scores reflect the degree of overlap between the predicted and ground truth object bounding box. As outputs, the model simultaneously predicts bounding boxes for classes of object and the confidence that an object is in the boxes.

Data preparation

To train YOLOv3 on DAS data, a training data set of example images of microseismic events and background noise is prepared. The synthetic data are prepared as described above in the "Data" section and in Baird et al. (2020), and 2000 microseismic event examples with a single event in each file are created. Two thousand examples were used to gain a good distribution of event magnitudes, locations, and source mechanisms while restricting the num-

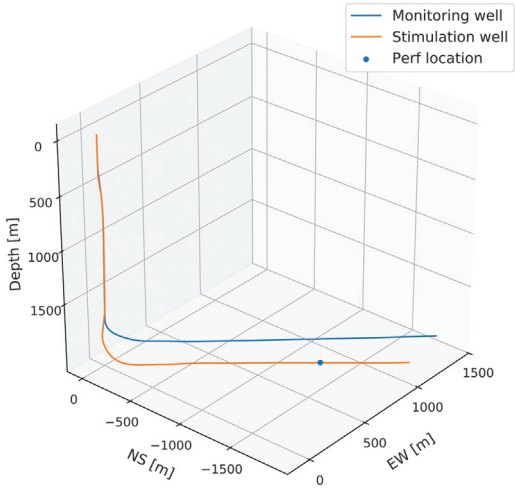

Figure 2. Schematic of the stimulation and monitoring well configuration for Data Set 1 and the synthetic data. A single fiber-optic cable is deployed in the monitoring well. Data Set 1 is recorded over one hydraulic fracturing stage. The location of the perforation shot for this stage is indicated by the blue dot.

ber to minimize training times. Only data from the horizontal section of the cable are used because very little microseismic energy actually was recorded on the vertical section in the field data due to the distance of the hydraulic fracturing from the build section of the well (>1 km). To reduce the size of the input images for the machine learning, data from every fourth channel is used. Downsampling rather than decimation using low-pass filtering was used for simplicity following initial tests that showed it did not affect the detection capabilities. The downsampling maintains sufficient detail in the waveforms while allowing data from the horizontal section of the cable to be included. Gray-scale images (512 × 512 pixels in size) are created (i.e., including data from a 2 km fiber section over 512 time samples, 0.256 s). 512 × 512 was chosen because it is the minimum size that allows input images to be used without incurring any time decimation. Initial tests also showed convergence of the training with this size of image. The longer processing times associated with larger images work against the aim to provide (near) real-time data processing.

To make the synthetic examples more realistic, examples of noise from the first minutes of the field data, in a random channel order, are added to the data with signal-to-noise ratios (S/Ns) between 3 and 7. Gaussian noise then is added at a random level between 0% and 25%. Thus, 2000 synthetic microseismic event images are created. The images are gray scale and hence have amplitudes normalized between 0 and 255. Ground truth bounding boxes for the events are automatically calculated because the location of the event, and hence the closest channel, is known. The box is chosen to include the first P- and S-wave arrivals on the closest channels. Some variation in size is introduced to replicate human variations in choosing a box and also because the extent of a microseismic event is somewhat subjective (how many channels should be included?). These automatically determined bounding boxes are the "ground truth" boxes, and predictions made by YOLOv3 are compared with these.

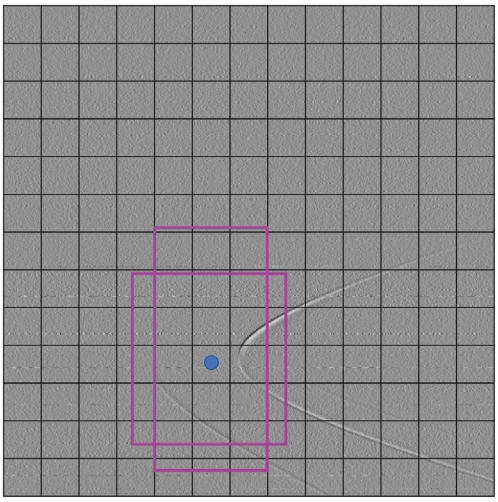

Figure 3. An example of bounding box testing by YOLOv3. The image shows a synthetic microseismic event recording with noise added. The image is divided into the grid shown and many bounding boxes. In this example, two boxes are tested for the existence of a microseismic event for each grid cell — here, for the grid cell indicated by the blue dot.

Two thousand example images of noise also are created in a similar way. In addition, lines of random length and orientation are added to half of the negative images. This is done with the aim of teaching the network that lines on images do not necessarily represent microseismic events. Such lines might represent, for example: noise spikes, faulty channels, or tube waves traveling along the array.

The data (events and noise) were not augmented in some ways that often are used in machine learning because, for example, rotation of an image of a microseismic event will not represent actual data recordings. However, the saturation and brightness of the image were changed at random, and the size and aspect ratio of the images were allowed to vary between 0.4 and 1.6 times the original. The network size also was allowed to vary.

The 4000 synthetic images are randomly split into training and validation data sets, with 3600 images chosen for training and the remaining 400 images (10% of the images) used for validation. No preprocessing was performed on the training data, and raw images were used in training.

Network training

YOLOv3 has been pretrained on ImageNet images. The network therefore was not trained from scratch but was retrained on 3600 synthetic example images of microseismic events and noise recordings,

Table 1. YOLOv3 input parameters used in training the network to detect microseismic events.

Run	Learning rate	Burn-in	Steps	Batch size
1	0.01	500	1600, 1800	64
2	0.01	1000	1600, 1800	64
3	0.001	500	1600, 1800	64
4	0.001	1000	1600, 1800	64
5	0.0001	1000	1600, 1800	64
6	0.001	1000	1600, 1800	16
7	0.0001	500	1600, 1800	64
8	0.001	500	1000, 1500	64
9	0.001	1000	1200, 1600	64

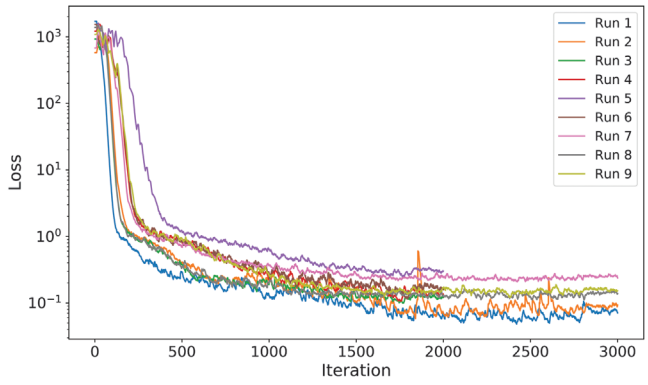

Figure 4. Model loss during training of YOLOv3 on 3600 synthetic data images. The parameters used in each training run are given in Table 1.

DOI:10.1190/geo2019-0774.1

using the weights from the pretrained model to initialize the training. This reduces the training time required. The aim is to find the parameters that give the best event detection results while minimizing the number of missed events and false detections. The learning rate determines how quickly the model can adapt in training. In YOLOv3, the learning can be modified by several parameters and a learning rate scheduler is used, determined by the burn-in rate. For the initial burn-in (BI) iterations, the learning rate for that iteration (I) is given by

$$\mathrm{LR}(I) = \mathrm{LR} \times (I/\mathrm{BI})^4 \quad (1)$$

where LR is the learning rate defined by the user that is used after BI iterations. Later in the training, the weights should not vary significantly between iterations so a smaller learning rate can be used, and LR is multiplied by 0.1 at the given step iterations. LR values between 0.01 and 0.0001 are tested, and $0.001 \leq \mathrm{LR} \leq 0.01$ produce lower losses. A selection of learning rate scheduler parameters and batch sizes tested are given in Table 1.

Initially, the network size is set to the pixel size of the training images (512×512) to preserve details in the waveforms. Time decimation affects the accuracy of first-arrival identification, and through initial testing it was determined that lower losses were achieved if the network size allowed input without any downsampling in time. The width and height of the network are set to the same size and allowed to vary in the training every 10 iterations. Batch sizes, or number of images passed through the network in one iteration, between 16 and 128 were tested. If the number is too large, the training becomes slow. However, if a very small number is used, the network may not experience a good sample of the range of object attributes and the results could vary wildly between each iteration. The maximum batch size allowing the network training to run in a few hours was 64, and this is the value used for most of the results presented here (Table 1).

Network losses from the selection of training tests in Table 1 are shown in Figure 4. These results include the training runs with the smallest losses, and the weights from the three training attempts with the lowest loss are tested on the validation set (Runs 1, 2, and 3 in Table 1 and Figure 4). It is observed that, after 2000 iterations, the loss stabilizes; therefore, the weights from the 2000th iteration are used in the validation to avoid overtraining the network.

Validation

Batch image detection (Gong, 2019) is conducted using the weights from the trained networks from Runs 1, 2, and 3 (Table 1). The validation images to which this is applied comprise 194 images containing microseismic events (one event per image) and 206 noise images. The outputs from this detection are bounding boxes containing microseismic events with a given confidence. A minimum confidence of 25% is used to declare the detection of an event. Detections are classified as either true positives (the network is detecting a microseismic event) or false positives (an event is declared in which there is no seismic event). Images in which no event is detected with >25% confidence may be true negatives (it is an image containing noise only) or false negatives (the image contains an undetected seismic event).

The classification results using the weights from Runs 1, 2, and 3 are shown in Table 2. All models correctly identify >95% of the events in the data set, and >99% of the noise images are classified as such. It also is important not to falsely report events, and in each

Table 2. Microseismic event classification results for YOLOv3 model validation using weights from Runs 1, 2, and 3.

	Run 1	Run 2	Run 3
True positives	191	190	187
True negatives	204	206	206
False positives	4	1	2
False negatives	3	4	7

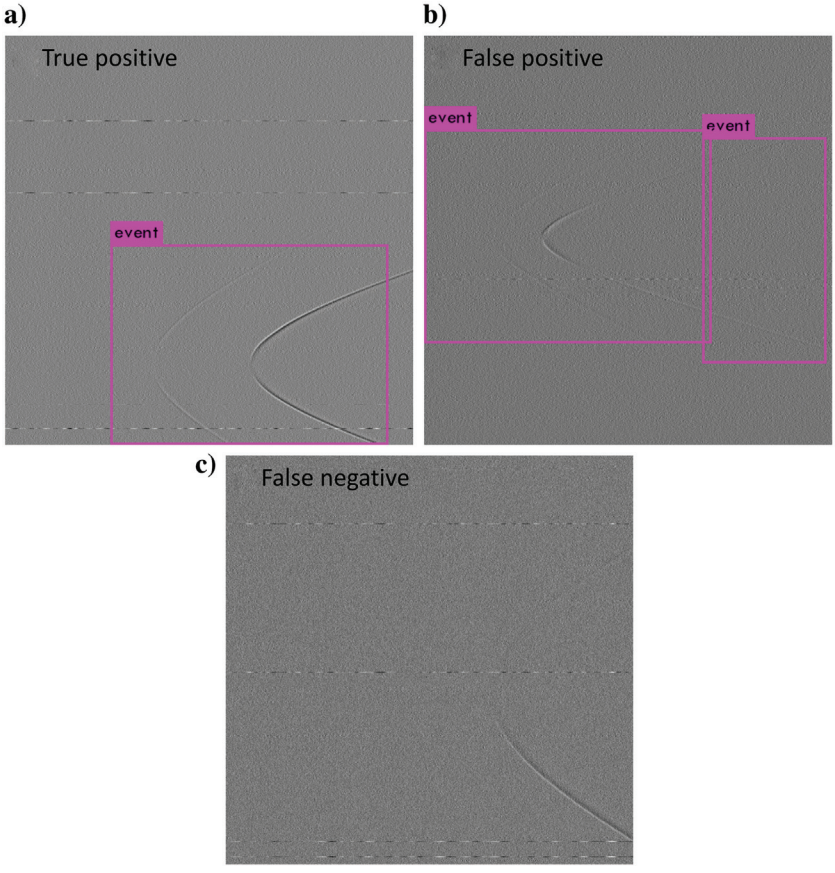

Figure 5. Example event detections from the YOLOv3 model validation. One example of (a) a true-positive event detection, (b) a false-positive event detection (the right-side box), and (c) an image falsely classified as noise are given.

of the three cases, ≤ 2% of detections are false positives. Figure 5 shows examples of true-positive, false-positive, and false-negative event detection. In total, eight events were missed in one or more of the validation tests. The dominant causes of missed detections are the event mechanism and location effects on the features observed in the waveforms. Four out of eight of the events had one-sided radiation patterns; that is, arrivals were obvious on only one side of the apex (e.g., Figure 5c), and for six out of eight events, only S-waves, not P-waves, are visible (e.g., Figure 5c). Two of the missed events have first arrivals in the latter 15% of the image so the full event is not visible, and two have high-amplitude noise on three or more channels covering the event signal. Training with more images with these characteristics could improve detection accuracy. The threshold for event detection is set low (at 25%). However, in all three cases, >70% of events are detected with a confidence >80% with Runs 1 and 2 providing the highest confidence detections (Figure 6).

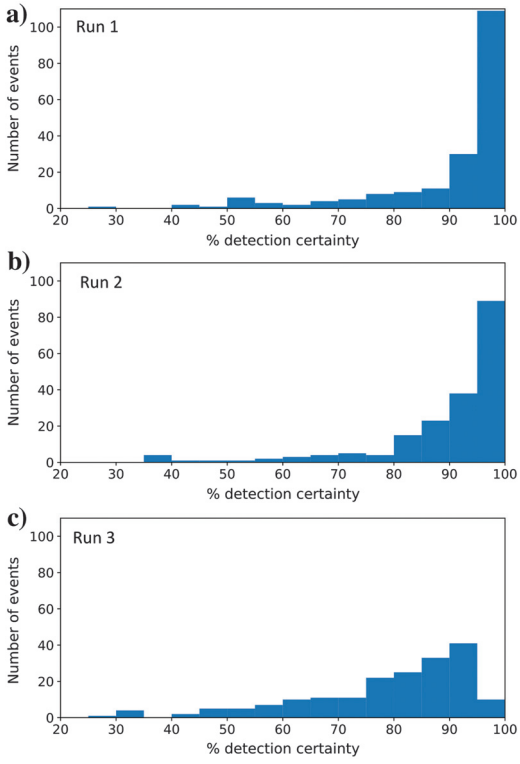

Figure 6. Histograms of microseismic event detection certainty in Data Set 1 for (a) Run 1, (b) Run 2, and (c) Run 3. The YOLOv3 input parameters are given in Table 1.

Testing with field data

We test the capabilities of the trained CNN to detect events in field data because, obviously, to be useful, the method must be applicable to real data. Eight hundred images are created from the Data Set 1 DAS data. These images were manually inspected without any filtering applied, and 527 events were observed.

In validation, the weights from Run 2 (with the parameters in Table 1) produce slightly better results, with more detected events and fewer missed events than the other validation tests. Therefore, we use the trained YOLOv3 network with these weights to test detection in Data Set 1, and >450 events are detected in the 800 test images (Data Set 1 in Table 3).

The machine-learning approach is able to detect approximately 80% of the number found by manual inspection. Encouragingly, the network sometimes is able to distinguish some overlapping events (e.g., Figure 7a) even though the synthetic training images contained only one event per image. This multievent detection is vital in an industrial environment in which microseismic events often

Table 3. YOLOv3 microseismic event detection on field data using the weights from Run 2.

Data set	1	2	3	1	1
Preprocessing		None		f-k filter	Downsampled
True events	457	126	13	517	242
False detections	10	2	2	27	13
Missed events	70	47	11	50	267

The results of event detection on raw data from three different data sets are shown. Data Set 1 consists of 800 images. Data Set 2 is 1 min of data, and Data Set 3 is 6 s of data. In addition, detection results are shown for filtered Data Set 1 data and Data Set 1 data downsampled by a factor of 2 in space and time.

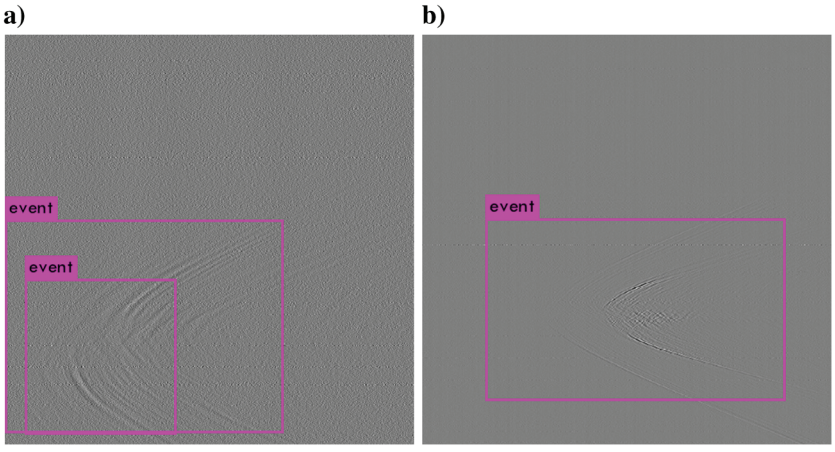

Figure 7. YOLOv3 microseismic event detection in field data: (a) Two overlapping events (indicated by the pink boxes) are distinguished by the network in Data Set 1 and (b) an example event detection in Data Set 2.

occur in quick succession or simultaneously at different locations. Only 10 false detections are declared — 2% of the events or one false detection every 20 s. This indicates that the network has been well-trained to distinguish noise from events.

Most of the events missed in the detection with the CNN have visible P and S arrivals on both sides on the apex without obvious channel noise or first arrivals at the edge of the image. Therefore, the characteristics of the missed events are different from the events missed in the validation set of images. The dominant characteristic is the S/N compared with the detected events. The maximum values of individual channel STA/LTA characteristic functions of the detected and undetected events are shown in Figure 8. Some 90% of the missed events have maximum values <7.5, whereas for the detected events the proportion is 50%. Lower characteristic function values make it less likely that an event is detected, although the S/N is likely not the only feature important in detection. The magnitude, mechanisms, and locations of the events are not reliably known and therefore cannot be investigated thoroughly.

It is essential that it is possible to generalize the application of machine-learning approaches, so we also test the network's detection capabilities on other data sets, namely, Data Sets 2 and 3. These are data sets with velocity models and well configurations previously unseen by the model. Additionally, Data Set 2 is recorded on the Silixa Carina system rather than the iDAS v2 system used to record Data Sets 1 and 3. Data Set 2 is 1 min of data recorded on an "L-shaped" (vertical and horizontal) well, and Data Set 3 consists of only 6.0 s of data recorded in a horizontal well. In Data Set 2, 126 events are detected with the machine-learning algorithm, >70% of events identified by manual inspection, and an example is shown in Figure 7b. In Data Set 3, >50% of manually identified events are detected (Table 3). The fact that the trained CNN is able to detect events in data from a previously unseen setting is vital for the general application of a machine-learning approach to microseismic event detection. The results confirm the ability of the trained CNN to detect events in DAS data recorded on different cable configurations and by different recording systems in varying geologic settings.

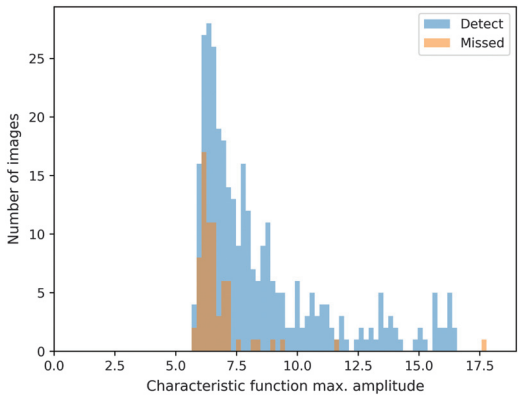

Figure 8. Histograms of STA/LTA characteristic function maximum values for detected and undetected (missed) events in the Data Set 1 test images. Lower maximum values indicate lower S/N recordings.

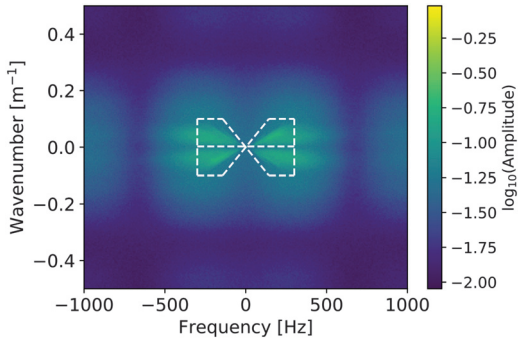

Figure 9. The summed f-k spectrum of 100 test events. The white dashed lines outline the f-k filter chosen to conduct microseismic event detection.

EVENT DETECTION WITH 2D FILTERING AND DATA STACKING

In addition to a machine-learning approach, we also test classic signal processing and detection methods for event detection in DAS data. This is done for comparison with the machine-learning approach. For this processing stream, we concentrate our efforts to improve the S/N of recordings using techniques applied in image processing by considering the data as a 2D array. This is because the large number of recording channels and the high sampling rate used to acquire DAS data produce large data volumes. If each trace is treated individually, as is commonly the case for geophone arrays, processing the data from DAS arrays results in significantly longer times.

It is possible to treat DAS data as a 2D array (or image) because the wavefield is well-sampled in space and time. This presents a range of filtering options to enhance object features, depending on the types of features of interest. In a data processing context, they are available in packages such as scikit-image (van der Walt et al., 2014).

A 2D median filter is initially identified as a suitable method to improve S/Ns in the DAS data. The 2D median filtering method is a noise-reduction technique and a nonlinear low-pass filter

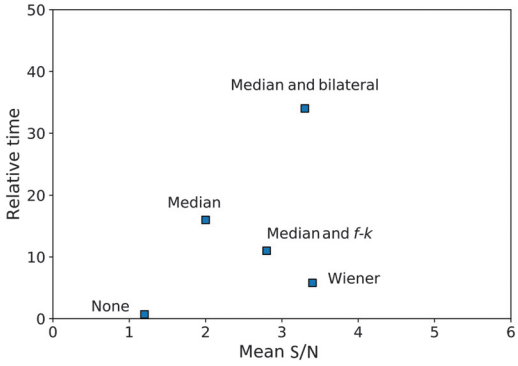

Figure 10. The relative time required to process 100 microseismic event files with (a) the median amplitude removed (labeled "None") and using (b) a 2D median filter with a local window size of 5, (c) a Wiener filter with a filter window size of 11, (d) a 2D median filter (window size = 3) and a bilateral filter with a window size of 5, and (f) a 2D median filter (window size = 3) and an f-k filter with 0.1 Hz < f < 300 Hz, 0.0025m^{-1} < k <0.1m^{-1} and a maximum slowness of 0.0007 s/m. The results are shown as a function of the mean S/N calculated for the 100 events.

DOI:10.1190/geo2019-0774.1

(e.g., Huang et al., 1979). A pixel value is replaced by the median value of a square of pixels surrounding it. It is useful when an image contains low to medium levels of Gaussian noise, as is often assumed in seismic data processing. A median filter better preserves the edges of features than a Gaussian filter. The onset (edge) of a seismic wave is an important parameter in event detection; therefore, a 2D median filter is applied to the data rather than a Gaussian filter. More sophisticated adaptive median filters are available (e.g., Hwang and Haddad, 1995) but these inevitably further increase the processing time.

Several other candidate filtering methods were tested, including Wiener and bilateral filtering. Bilateral and Wiener filters are routinely applied in image processing. A Wiener filter is a type of deblurring filter, and a bilateral filter is an edge-preserving filter that, rather than calculating the median value of the surrounding pixels, computes a weighted average of surrounding pixel intensities. The weights are based on distance and intensity similarity. For details on bilateral filtering, see Paris et al. (2008).

Frequency-wavenumber (f-k) filtering often is performed in seismic processing to improve S/N, so this method is tested on the DAS data. To perform the f-k filtering, the time/space domain signal is transformed to the frequency/wavenumber domain using a 2D fast Fourier transform. Noise with different frequencies and wavenumbers than the signal of interest can be masked before transforming the signal back to the time/space domain. The f-k filter applied in this study is developed specifically for application to this DAS data set based on the recording parameters and the frequency content of 100 example events. To determine the frequency content of the signal and the range of apparent velocities of the waves detected, the f-k spectra of the example events are summed (Figure 9). A tapered mask with a Gaussian filter is applied to the data before an inverse Fourier transform is applied. Masks with higher maximum frequencies, >300 Hz, were tested, but fewer events were detected. Generally, if example data are unavailable, a filter can be developed with the expected maximum frequency and apparent seismic velocities estimated from the expected event locations relative to the monitoring array and the velocity model.

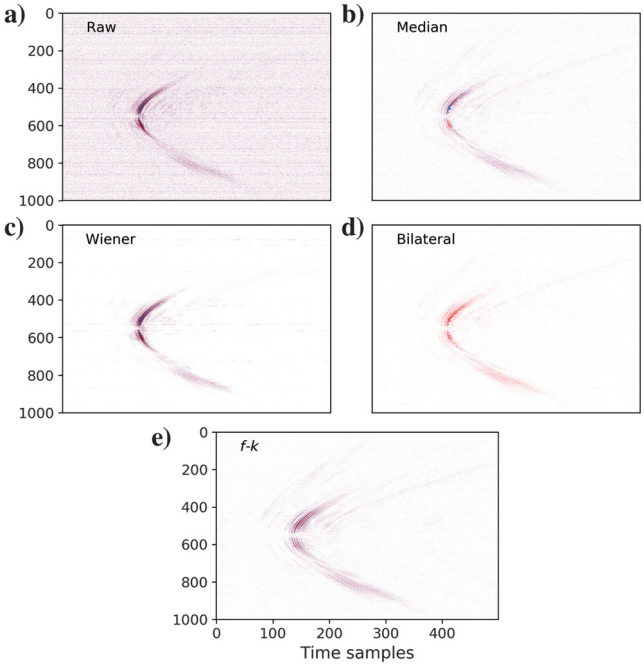

Figure 11. Example microseismic event recording (a) with no filtering, (b) 2D median filtered with a local window size of 5; (c) Wiener filtered with a filter window size of 11, (d) 2D median filtered (window size = 3) and a bilateral filter applied with a window size of 5, and (e) 2D median filtered (window size = 3) and an f-k filter applied.

The different filtering methods are compared by calculating the average S/N improvements for 100 microseismic events from Data Set 1. The S/N is defined as the average root mean square (rms) amplitude of the 100 samples on all channels after the detection time divided by the rms amplitude of 100 time samples before the detection. Due to the distance of the events from the cable, the window includes P- and S-wave arrivals on the closest channels. Bilateral, Wiener, and f-k filters are found to result in good increases in the overall S/N (Figure 10). To enable detection in near real time, the bilateral filter is discounted. The filter window lengths required to achieve good S/Ns with the Wiener filter also result in blurring of the data; this reduces the onset sharpness and S/N of the first arrival (Figure 11). To encourage event detection on P- rather than S-waves, the f-k filter combined with a 2D median filter is chosen as the preferred filter for event detection. This filter also better preserves characteristics of refracted and reflected waves and the location of polarity flips that may be useful in imaging or focal mechanism studies.

To test event detection with classic methods, the filtering is performed in two stages: First, a 2D median filter is applied, followed by 2D Fourier filtering. The only preprocessing of the data is removal of the mean amplitude to give zero-mean data.

We make use of the large data volumes by summing (or stacking) the filtered data recorded over a range of channels. Data stacking is a technique that often is used in microseismic event detection, particularly for use with surface arrays (e.g., Chambers et al., 2010). Here, initially a characteristic function is calculated for each channel using a recursive STA/LTA algorithm (Allen, 1978). Then, the average amplitude sum of the characteristic functions is computed over each of the N channels for each time sample:

$$A(t) = \frac{1}{N}\sum_{i=0}^{N} |a_i(t)|. \tag{2}$$

Linear stacking is chosen because it is the simplest method to sum data to improve S/N. Other methods (n^{th} root, semblance weighted, and instantaneous phase-weighted stacking) are tested for comparison (e.g., Figure 12). The phase-weighted stack does not produce distinguishable peaks where the events occur. Following stacking, an event is declared if this sum exceeds a given threshold. Here, we use 15% above the median background value (Figure 13).

The number of events detected in Data Set 1 (4 h of field data) using all of the stacking methods is given in Table 4, and the largest number of events is detected with the linear stack. This method also has the best success rate (72% of the detections are true events). The success and simplicity (and therefore the speed) of the linear stacking make this the favored method for event detection. Using a 2D median and f-k filter followed by linear stacking of STA/LTA characteristic functions, >1900 events are detected in total in Data Set 1.

Comparing the detection capabilities of the trained CNN and the chosen f-k filtering method on the 800 test images from Data Set 1,

400 events are detected using the f-k filtering and stacking method compared with the 457 events detected using the trained CNN and the 527 detected in the unfiltered images by manual inspection (Table 3). Approximately 14% more events are found using the CNN than using the filtering and stacking method.

DISCUSSION

A current disadvantage of the use of DAS for seismic monitoring is that large volumes of data are produced. Accurate and near-real-time processing methods are required. This paper presents a proof of concept that a CNN can be trained to accurately detect seismic events in DAS data and that this network can be applied to detect events recorded on any cable deployed in a horizontal well. Further development is required to optimize the process for near-real-time data processing. Here, we present the benefits and limitations of the current model and outline the developments required to make this method applicable to any DAS microseismic monitoring situation.

In the present study, the YOLOv3 network is retrained using synthetic data. Synthetic data were chosen rather than real data for the training because they provide a data set in which the answer is known and the performance of the network can therefore be reliably validated. Initially, Gaussian noise was added to the synthetic waveforms, but in testing the network was unable to detect a significant number of events in real data. Further tests can be conducted on using different types of noise and real noise recorded in a variety of settings to improve detection performance. The results also highlight that the YOLOv3 model can be trained on a small number of examples (4000) compared with the tens of thousands or hundreds of thousands used in other published seismic event detection CNN training examples (e.g., Binder and Chakraborty, 2019; Zhu and Beroza, 2019). This reduces the required training time.

An advantage of CNN machine-learning techniques is that the network can be retrained as more data become available. As microseismic events are detected, these can be added to the training data set to improve the reliability of future detection. However, this can also be mimicked by adding further synthetic examples using different well configurations and geologic settings and structures. For example, synthetic data sets for vertical and "L-shaped" wells with different velocity models could be added to train the network to be better able to recognize events recorded on any cable configuration. It was found that the trained CNN is more likely to miss events recorded with low S/N; therefore, further examples with S/Ns of < 3 should be added to the training data. Synthetic training data sets can provide waveforms for events of any magnitude, mechanism, or location, and any number of noise examples or any S/N recordings can be created. Therefore, future detection should not be restricted to the types of events previously observed in a particular setting, as demonstrated by the initial success of event detection in Data Sets 2 and 3 in this study.

Images of the raw data are produced here for event recognition testing. The elimination of the need to conduct any preprocessing, due to the network having learned the best-performing filters, speeds up event recognition. To verify the effectiveness of the filters used by YOLOv3, the network also is tested on images produced from data filtered using the f-k filter described in the Event Detection with Machine Learning section. This increases the number of events detected by approximately 10% to 517 events (Table 3), indicating that the internal filters of the trained network are not yet optimized. If the network were trained on images with lower S/Ns (here, a minimum S/N of 3 was used to ensure that events could be manually identified), it should detect lower S/N signals; therefore,

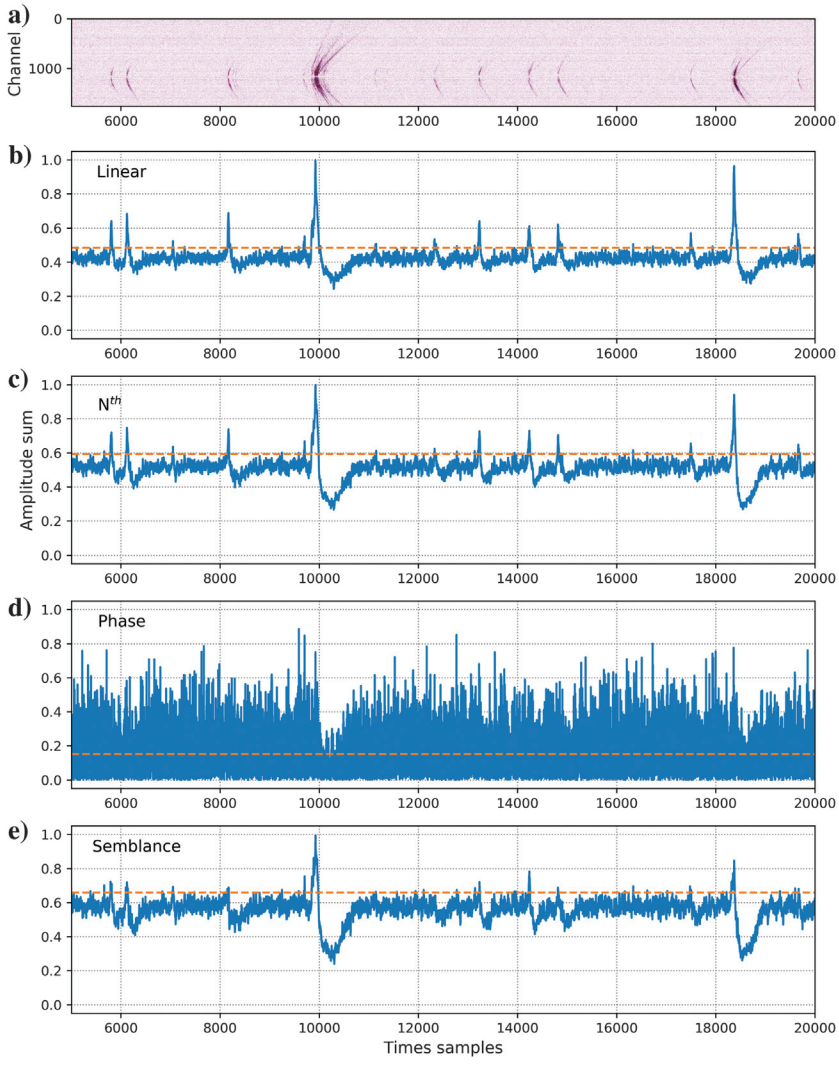

Figure 12. An example of the data stacking results for (a) a sample of data containing several microseismic events for (b) linear, (c) fourth root, (d) instantaneous phase-weighted, and (e) semblance-weighted stacks of the STA/LTA characteristic functions. The dashed lines indicate the detection threshold that are used (15% above the median value).

DOI:10.1190/geo2019-0774.1

the improvement observed with the f-k filtering could be eliminated.

Despite the endless possible architectures for CNN models, the YOLOv3 architecture stands out as solving the problem of real-time object detection in video data with competitive accuracy (~70%); therefore, it was chosen for this study of continuous DAS data. Detection accuracy of >80% is achieved in Data Set 1, indicating that the network is performing well. Significant factors in the speed of CNN training and object detection are not only the computer and the type of processor (CPU or GPU) used, but also the number and size of the input images/video. Several tutorials are available online on how to implement YOLOv3 effectively, and the GitHub repositories (Redmon, 2013) and (AlexeyAB, 2017) provide useful information. Using the trained network, we investigate whether the size of test images can be reduced while maintaining event detection capabilities. Any downsampling in time results in poorer detection performance, whereas spatial sampling can be reduced by a factor of 2 without affecting the performance. Downsampling in time and space so every other channel and time sample are used reduces the number of events detected by nearly half in Data Set 1 (Table 3). If the network were trained on time downsampled data, it is possible that this result could be improved.

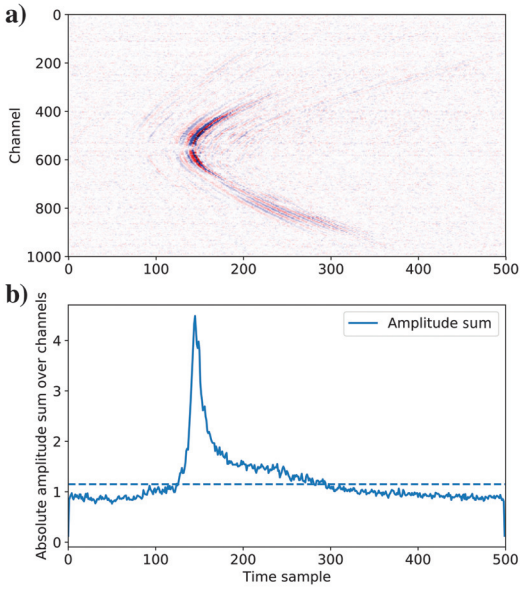

Figure 13. (a) An example microseismic event recorded on the DAS cable in the monitoring well. First-arrival P-waves and subsequent S-waves are observed. (b) The absolute amplitude sum for each channel over time. The dashed line indicates the threshold value for declaring the onset of an event.

Table 4. Number of microseismic events detected in Data Set 1 using different data stacking methods.

Method	Total number of events	% true detections
Linear	1960	72
Fourth root	1635	69
Semblance-weighted	852	46

The characteristic function of the STA/LTA of each channel is the input data to the stack. The percentage of detections being true seismic events is shown.

CONCLUSION

There has been a recent rise in the deployment of DAS fiber-optic monitoring for microseismic applications. This type of monitoring produces large data volumes; therefore, new processing techniques are required to enable data processing in near real time. This study presents the first successful application of a trained CNN to seismic event detection in multiple DAS data sets. We train YOLOv3 for object detection using a synthetic microseismic data set with real noise added, and we subsequently apply it to three field data sets. This is compared to more traditional event detection techniques. The data are processed with f-k filtering combined with a 2D median filter. Subsequently, a simple threshold detection algorithm is applied to a linear amplitude sum of the recursive STA/LTA characteristic functions of all channels.

In field data with the same well setup and geologic setting used to create the synthetics, the CNN is able to detect >80% of events of the 527 events found by manual inspection of unfiltered images. Some 400 of the events are detected with the filtering and threshold detection method. Importantly, the false detection rate of the CNN is low, 2% of detections. Lower S/N events are more likely to be missed in the detection. This effect could be reduced by training with lower S/N examples. In other data sets, in which the geologic setting, operational parameters, and the treatment-to-monitoring well distances are different and unknown, the CNN correctly identifies >50% of events. The application of CNNs to seismic event detection in DAS data is attractive due to the automatic nature, lack of preprocessing required, and potential speed of the technique. Detection accuracy could be improved in the future with the addition of further noise and event examples to the training process. The wider the variety of recording settings and S/Ns that are used for training, the more accurate the detection should become.

ACKNOWLEDGMENTS

The authors thank Chevron ETC and Silixa Ltd. for permission to publish the work. A. L. Stork and A. F. Baird are supported by NERC grant NE/R014531/1. We thank V. Gong for his help and advice on the batch image processing.

DATA AND MATERIALS AVAILABILITY

Data associated with this research are confidential and cannot be released.

APPENDIX A

YOLOV3 NETWORK ARCHITECTURE

Layer	Filters	Size	Input		Output
0	conv	32 3 x 3 / 1	512 x 512 x 3	->	512 x 512 x 32
1	conv	64 3 x 3 / 2	512 x 512 x 32	->	256 x 256 x 64
2	conv	32 1 x 1 / 1	256 x 256 x 64	->	256 x 256 x 32
3	conv	64 3 x 3 / 1	256 x 256 x 32	->	256 x 256 x 64
4	res	1	256 x 256 x 64	->	256 x 256 x 64
5	conv	128 3 x 3 / 2	256 x 256 x 64	->	128 x 128 x 128
6	conv	64 1 x 1 / 1	128 x 128 x 128	->	128 x 128 x 64
7	conv	128 3 x 3 / 1	128 x 128 x 64	->	128 x 128 x 128
8	res	5	128 x 128 x 128	->	128 x 128 x 128
9	conv	64 1 x 1 / 1	128 x 128 x 128	->	128 x 128 x 64
10	conv	128 3 x 3 / 1	128 x 128 x 64	->	128 x 128 x 128
11	res	8	128 x 128 x 128	->	128 x 128 x 128
12	conv	256 3 x 3 / 2	128 x 128 x 128	->	64 x 64 x 256
13	conv	128 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 128
14	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
15	res	12	64 x 64 x 256	->	64 x 64 x 256
16	conv	128 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 128
17	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
18	res	15	64 x 64 x 256	->	64 x 64 x 256
19	conv	128 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 128
20	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
21	res	18	64 x 64 x 256	->	64 x 64 x 256
22	conv	128 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 128
23	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
24	res	21	64 x 64 x 256	->	64 x 64 x 256
25	conv	128 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 128
26	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
27	res	24	64 x 64 x 256	->	64 x 64 x 256
28	conv	128 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 128
29	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
30	res	27	64 x 64 x 256	->	64 x 64 x 256
31	conv	128 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 128
32	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
33	res	30	64 x 64 x 256	->	64 x 64 x 256
34	conv	128 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 128
35	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
36	res	33	64 x 64 x 256	->	64 x 64 x 256
37	conv	512 3 x 3 / 2	64 x 64 x 256	->	32 x 32 x 512
38	conv	256 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 256
39	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
40	res	37	32 x 32 x 512	->	32 x 32 x 512
41	conv	256 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 256
42	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
43	res	40	32 x 32 x 512	->	32 x 32 x 512
44	conv	256 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 256
45	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
46	res	43	32 x 32 x 512	->	32 x 32 x 512
47	conv	256 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 256
48	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
49	res	46	32 x 32 x 512	->	32 x 32 x 512
50	conv	256 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 256
51	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
52	res	49	32 x 32 x 512	->	32 x 32 x 512
53	conv	256 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 256
54	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
55	res	52	32 x 32 x 512	->	32 x 32 x 512
56	conv	256 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 256
57	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
58	res	55	32 x 32 x 512	->	32 x 32 x 512
59	conv	256 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 256
60	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
61	res	58	32 x 32 x 512	->	32 x 32 x 512
62	conv	1024 3 x 3 / 2	32 x 32 x 512	->	16 x 16 x1024
63	conv	512 1 x 1 / 1	16 x 16 x1024	->	16 x 16 x 512
64	conv	1024 3 x 3 / 1	16 x 16 x 512	->	16 x 16 x1024
65	res	62	16 x 16 x1024	->	16 x 16 x1024
66	conv	512 1 x 1 / 1	16 x 16 x1024	->	16 x 16 x 512
67	conv	1024 3 x 3 / 1	16 x 16 x 512	->	16 x 16 x1024
68	res	65	16 x 16 x1024	->	16 x 16 x1024
69	conv	512 1 x 1 / 1	16 x 16 x1024	->	16 x 16 x 512
70	conv	1024 3 x 3 / 1	16 x 16 x 512	->	16 x 16 x1024
71	res	68	16 x 16 x1024	->	16 x 16 x1024
72	conv	512 1 x 1 / 1	16 x 16 x1024	->	16 x 16 x 512
73	conv	1024 3 x 3 / 1	16 x 16 x 512	->	16 x 16 x1024
74	res	71	16 x 16 x1024	->	16 x 16 x1024
75	conv	12 1 x 1 / 1	16 x 16 x1024	->	16 x 16 x 512
76	conv	1024 3 x 3 / 1	16 x 16 x 512	->	16 x 16 x1024
77	conv	512 1 x 1 / 1	16 x 16 x1024	->	16 x 16 x 512
78	conv	1024 3 x 3 / 1	16 x 16 x 512	->	16 x 16 x1024
79	conv	512 1 x 1 / 1	16 x 16 x1024	->	16 x 16 x 512
80	conv	1024 3 x 3 / 1	16 x 16 x 512	->	16 x 16 x1024
81	conv	18 1 x 1 / 1	16 x 16 x1024	->	16 x 16 x 18
82	yolo				
83	route	79			
84	conv	256 1 x 1 / 1	16 x 16 x 512	->	16 x 16 x 256
85	upsample	2x	16 x 16 x 256	->	32 x 32 x 256
86	route	85 61			
87	conv	256 1 x 1 / 1	32 x 32 x 768	->	32 x 32 x 256
88	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
89	conv	256 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 256
90	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
91	conv	256 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 256
92	conv	512 3 x 3 / 1	32 x 32 x 256	->	32 x 32 x 512
93	conv	18 1 x 1 / 1	32 x 32 x 512	->	32 x 32 x 18
94	yolo				
95	route	91			
96	conv	128 1 x 1 / 1	32 x 32 x 256	->	32 x 32 x 128
97	upsample	2x	32 x 32 x 128	->	64 x 64 x 128
98	route	97 36			
99	conv	128 1 x 1 / 1	64 x 64 x 384	->	64 x 64 x 128
100	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
101	conv	128 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 128
102	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
103	conv	128 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 128
104	conv	256 3 x 3 / 1	64 x 64 x 128	->	64 x 64 x 256
105	conv	18 1 x 1 / 1	64 x 64 x 256	->	64 x 64 x 18
106	yolo				

Figure A-1. YOLOv3 network architecture.

REFERENCES

AlexeyAB, 2018, AlexeyAB_darknet, Last accessed February 2020 at https://github.com/AlexeyAB/darknet/issues/1695#issuecomment-450995001, GitHub repository.

AlexeyAB, 2017, AlexeyAB_darknet, Last accessed July 2020 at https://github.com/AlexeyAB/darknet, GitHub repository.

Allen, R. V., 1978, Automatic earthquake recognition and timing from single traces: Bulletin of the Seismological Society of America, **68**, 1521–1532.

Baird, A. F., A. L. Stork, S. A. Horne, G. Naldrett, J.-M. Kendall, J. Wookey, J. P. Verdon, and A. Clarke, 2020, Characteristics of microseismic data recorded by distributed acoustic sensing (DAS) systems in anisotropic media: Geophysics, **85**, no. 4, KS139–KS147, doi: 10.1190/geo2019-0776.1.

Beresnev, I. A., and G. M. Atkinson, 1997, Modeling finite-fault radiation from the ω^n spectrum: Bulletin of the Seismological Society of America, **87**, 67–84.

Binder, G., and D. Chakraborty, 2019, Detecting microseismic events in downhole distributed acoustic sensing data using convolutional neural networks: SEG 89th Annual International Meeting, Expanded Abstracts, 4864–4868, doi: 10.1190/segam2019-3214863.1.

Chambers, K., J.-M. Kendall, S. Brandsberg-Dahl, and J. Rueda, 2010, Testing the ability of surface arrays to monitor microseismic activity: Geophysical Prospecting, **58**, 821–830, doi: 10.1111/j.1365-2478.2010.00893.x.

Chapman, C., 2004, Fundamentals of seismic wave propagation: Cambridge University Press.

Clarke, H., J. Verdon, T. Kettlety, A. Baird, and J.-M. Kendall, 2019, Real time imaging, forecasting and management of human-induced seismicity at Preston New Road, Lancashire, England: Seismological Research Letters, **90**, 1902–1915, doi: 10.1785/0220190110.

Cole, S., M. Karrenbach, D. Kahn, J. Rich, K. Silver, and D. Langton, 2018, Source parameter estimation from DAS microseismic data: SEG 88th Annual International Meeting, Expanded Abstracts, 4928–4932, doi: 10.1190/segam2018-2995716.1.

Dou, S., N. Lindsey, A. M. Wagner, T. M. Daley, B. Freifeld, M. Robertson, J. Peterson, C. Ulrich, E. R. Martin, and J. B. Ajo-Franklin, 2017, Distributed acoustic sensing for seismic monitoring of the near surface: A traffic-noise interferometry case study: Scientific Reports, **7**, 11620, doi: 10.1038/s41598-017-11986-4.

Gong, V., 2019, VG_AlexeyAB_darknet, Last accessed September 2019 at https://github.com/vincentgong7/VGAlexeyABdarknet, GitHub repository

Hartog, A. H., 2017, An introduction to distributed optical fibre sensors: CRC Press.

Horne, S., A. Baird, A. Stork, and G. Naldrett, 2019, Machine learning for DAS microseismic event detection: 81st Annual International Conference and Exhibition, EAGE, Extended Abstracts, WS10_13, doi: 10.3997/2214-4609.201901972.

Huang, T. S., G. J. Yang, and G. Y. Tang, 1979, A fast two-dimensional median filtering algorithm: IEEE Transactions on Acoustics, Speech, and Signal Processing, **27**, 13–18, doi: 10.1109/TASSP.1979.1163188.

Hui, J., 2018, Object detection: Speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3), Last accessed February 2020 at https://medium.com/Ojonathamhui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359,.

Hwang, H., and R. A. Haddad, 1995, Adaptive median filters: New algorithms and results: IEEE Transactions on Image Processing, **4**, 499–502, doi: 10.1109/83.370679.

Jousset, P., T. Reinsch, T. Ryberg, H. Blanck, A. Clarke, R. Aghayev, G. Hersir, J. Henninges, M. Weber, and C. M. Krawczyk, 2018, Dynamic strain determination using fibre-optic cables allows imag-

ing of seismological and structural features: Nature Communication, **9**, 2509, doi: 10.1038/s41467-018-04860-y.

Karrenbach, M., S. Cole, A. Ridge, K. Boone, D. Kahn, J. Rich, K. Silver, and D. Langton, 2019, Fiber-optic distributed acoustic sensing of microseismicity, strain and temperature during hydraulic fracturing: Geophysics, **84**, no. 1, D11–D23, doi: 10.1190/geo2017-0396.1.

Miller, D. E., T. M. Daley, D. White, B. M. Freifeld, M. Robertson, J. Cocker, and M. Craven, 2016, Simultaneous acquisition of distributed acoustic sensing VSP with multi-mode and single-mode fiber-optic cables and 3C-geophones at the Aquistore CO_2 storage site: CSEG Recorder, **41**, 28–33.

Mondanos, M., and T. Coleman, 2019, Application of distributed fibre-optic sensing to geothermal reservoir characterization and monitoring: First Break, **37**, 51–56.

Ning, I. L. C., and P. Sava, 2018, Multicomponent distributed acoustic sensing: Concept and theory: Geophysics, **83**, no. 2, P1–P8, doi: 10.1190/geo2017-0327.1.

Paris, S., P. Kornprobst, J. Tumblin, and F. Durand, 2008, Bilateral filtering: Theory and applications: Foundations and Trends in Computer Graphics and Vision, **4**, 1–75, doi: 10.1561/0600000020.

Rawat, W., and Z. Wang, 2017, Deep convolutional neural networks for image classification: A comprehensive review: Neural Computation, **29**, 2352–2449, doi: 10.1162/neco_a_00990.

Redmon, J., and A. Farhadi, 2018, YOLOv3: An incremental improvement, Technical Report: arXiv, 10804.02767.

Redmon, J., 2013, darknet, Last accessed July 2020 at https://github.com/pjreddie/darknet, GitHub repository.

Ren, S., K. He, R. Girshick, and J. Sun, 2015, Faster R-CNN: Towards real-time object detection with region proposal networks, *in* C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., Advances in Neural Information Processing Systems 28: Neural Information Processing Systems Foundation Inc., 91–99, https://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks, accessed 26 June 2020.

Ross, Z., M.-A. Meier, and E. Hauksson, 2018, P wave arrival picking and first-motion polarity determination with deep learning: Journal of Geophysical Research, **123**, 5120–5129, doi: 10.1029/2017JB015251.

Thomsen, L., 1986, Weak elastic anisotropy: Geophysics, **51**, 1954–1966, doi: 10.1190/1.1442051.

Stanford Vision Lab, 2016, ImageNet, Last accessed July 2020 at www.image-net.org.

van der, S., Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, and the scikit-image contributors, 2014, scikit-image: Image processing in python: PeerJ, **2**, e453, doi: 10.7717/peerj.453.

Vavryčuk, V., 2005, Focal mechanisms in anisotropic media: Geophysical Journal International, **161**, 334–346, doi: 10.1111/j.1365-246X.2005.02585.x.

Verdon, J. P., S. Horne, A. Clarke, A. L. Stork, A. F. Baird, and J.-M. Kendall, 2020, Microseismic monitoring using a fiber-optic distributed acoustic sensor array: Geophysics, **85**, no. 3, KS89–KS99, doi: 10.1190/geo2019-0752.1.

Webster, P., M. Molenaar, and C. Perkins, 2016, DAS microseismic: CSEG Recorder, **41**, 38–39.

Woollam, J., A. Rietbrock, A. Bueno, and S. D. Angelis, 2019, Convolutional neural network for seismic phase classification, performance demonstration over a local seismic network: Seismological Research Letters, **90**, 491–502, doi: 10.1785/0220180312.

Zhang, X., C. Yuan, J. Zhang, Z. C. S. Liu, and W. Li, 2018, Automatic microseismic detection and location via the deep convolutional neural network: SEG 88th Annual International Meeting, Expanded Abstracts, 3057–3061, doi: 10.1190/segam2018-2997548.1.

Zhou, Y., H. Yue, Q. Kong, and S. Zhou, 2019, Hybrid event detection and phase-picking algorithm using convolutional and recurrent neural networks: Seismological Research Letters, **90**, 1079–1087, doi: 10.1785/0220180319.

Zhu, W., and G. C. Beroza, 2019, Phasenet: A deep-neural-network-based seismic arrivaltime picking method: Geophysical Journal International, **216**, 1831–1841, doi: 10.1093/gji/ggy529.