GEOPHYSICS®

# A convolutional neural network approach to deblending seismic data



| | |
|---|---|
| Journal: | *Geophysics* |
| Manuscript ID | GEO-2019-0173.R2 |
| Manuscript Type: | Machine Learning and data analytics for geoscience applications |
| Keywords: | deblending, artificial intelligence, neural networks, signal processing |
| Area of Expertise: | Signal Processing |
| | |

# A convolutional neural network approach to deblending seismic data

Jing Sun[1,2,3], Sigmund Slang[2,3], Thomas Elboth[3], Thomas Larsen Greiner[2,4], Steven McDonald[3], Leiv-J Gelius[2]

Right Running head: A CNN approach to seismic deblending

[1]Jilin University, College of Geo-exploration Science and Technology, 130061 Changchun, China. Email: jingsun8803@hotmail.com.

[2]University of Oslo, Department of Geosciences, Sem Sælands vei 1, 0371 Oslo, Norway. Email: sigmund.slang@gmail.com, l.j.gelius@geo.uio.no.

[3]CGG. Email: Thomas.Elboth@CGG.com; stevenjmcdonald@hotmail.com.

[4]Lundin Norway AS, Strandveien 4, 1366 Lysaker, Norway. Email: tgreiner1983@gmail.com.

## ABSTRACT

For economic and efficiency reasons, blended acquisition of seismic data is becoming increasingly commonplace. Seismic deblending methods are computationally demanding and normally consist of multiple processing steps. Furthermore, the process of selecting parameters is not always trivial. Machine learning-based processing has the potential to significantly reduce processing time and to change the way seismic deblending is carried out. We present a data-driven deep learning-based method for fast and efficient seismic deblending. The blended data are sorted from the common source to the common channel domain to transform the character of the blending noise from coherent events to incoherent contributions. A convolutional neural network (CNN) is designed according to the special characteristics of seismic data, and performs deblending with results comparable to those obtained with conventional industry deblending algorithms. To ensure authenticity, the blending was done numerically and only field seismic data were employed, including more than 20000 training examples. After training and validating the network, seismic deblending can be performed in near real time. Experiments also show that the initial signal-to-noise ratio (SNR) is the major factor controlling the quality of the final deblended result. The network is also demonstrated to be robust and adaptive by using the trained model to firstly deblend a new data set from a different geological area with a slightly different delay time setting, and secondly to deblend shots with blending noise in the top part of the record.


## INTRODUCTION

In conventional seismic acquisition, a sufficiently large time interval between successive shot records is typically chosen to avoid the overlap of useful reflection events from

shot to shot. This implies that the source domain is often poorly sampled since the total number of shots needs to be kept at an acceptable minimum to reduce operational costs (Berkhout, 2008). To overcome such limitations in efficiency, the concept of blended acquisition has been introduced, where two or more shots are fired overlapping or almost simultaneously with time differences defined by a small random jitter (Barbier, 1982; Timoshin and Chizhik, 1982; Vaage, 2005; Beasley, 2008; Huo et al., 2009; Berkhout et al., 2010). To decompose the blended data into separate source contributions is a challenging task in seismic processing. Unlike many denoising problems, the coherent character of the blending noise closely resembles that of the signals to be recovered. Therefore, it may not be optimal to perform the deblending operation only on the source gathers directly.

In recent years, several attempts have been made to develop effective deblending techniques that aim to combine low computational cost with high data quality. Currently existing deblending methods fall into four categories: inversion-based methods, denoising-based methods, combinations of above two, and seismic apparition. A large number of algorithms have been proposed and we only mention a few here as examples.

The fundamental concept of inversion-based methods is to add appropriate constraint to the specific blending equation, and solve the inversion problem by inverting the matrix of the forward modeling operator or by using an iterative framework that iteratively estimates the useful signal and subtracts the blending noise. Berkhout (2008) proposed to apply a data-driven inversion of the blended records, and Neelmani et al. (2008, 2010) used a forward modeling approach to deblend the simultaneously acquired seismic data. An alternative inversion strategy was introduced by Herrmann et al. (2009), where the actual seismic deblending step

was carried out in the Curvelet domain. These approaches have been followed by a series of refinements of the iterative formulation (Mahdad et al., 2011, 2012; Doulgeris and Bube, 2012; Chen et al., 2014). Conversely, Wapenaar et al. (2012 a, b) suggested that the deblending of densely sampled sources can be implemented as a direct (i.e. non-iterative) inversion of the blending operator by taking the spatial band limitation into account.

Denoising-based methods make use of the nature of the random jitter and typically resort seismic data from the shot domain to another domain where contributions from nearly simultaneously fired sources are incoherent. Examples include sorting into the common channel, common offset, and common midpoint domain, often in combinations with data transforms to, for example, Wavelet (Chakraborty et al., 1995), Curvelet (Candes et al., 2006), Shearlet (Kutyniok and Lim, 2011), Seislet (Fomel et al., 2010) or Radon (Helgason, 1999) domain. In this way, the deblending process can be reformulated as the problem of removing incoherent noise (Moore et al., 2008; Akerberg et al., 2008; Maraschini, 2012; Chen, 2015). One feature common to all the methods discussed so far is the need to repeat the full set of calculation for each new blended data set.

Robertsson et al. (2016) proposed to reconstruct recorded interfering wavefields from two or more sources excited simultaneously by the principle of signal apparition, and extended it to separate seismic data acquired with multiple sources (Andersson et al., 2016). In blended acquisition, the amplitudes of the N and N+1 shots will be similar if they are fired nearly simultaneously, and the blending noise will appear in almost the entire record length of each shot. Seismic apparition tries to solve this problem by moving to the frequency-wavenumber (F-K) domain. However, when deploying a large number of sources, the sampling requirement

of seismic apparition is such that all the sources need to be fired very often. This makes it very difficult to maintain a reasonable source volume and to have time to recharge the individual air guns. Furthermore, the so-called flawless diamond of seismic apparition becomes increasingly small as we add sources. It only reaches around 10Hz for a hexa-source setup. These combined issues mean that seismic apparition style shooting is suboptimal when five-to-six simultaneous sources are deployed.

In this paper we propose an alternative processing path based on Machine Learning (ML). Even though the ML methods are computer intensive during the training process, once the network is fully trained the application of such methods can be carried out in near real time. According to LeCun et al. (2015), a disadvantage of conventional machine-learning techniques is the limitation in their ability to process natural data in their raw form. After normalization of such data, conventional ML techniques can perform better. Nevertheless, in applications such as computer vision related tasks, a specific class of techniques called deep learning is now more commonly used than conventional ML techniques.

Deep learning, as a ML technique, allows computational models that are composed of multiple processing layers to learn to represent data using multiple levels of abstraction (LeCun et al., 2015). Traditionally, this learning process has been achieved by the use of fully connected deep-layer artificial neural networks (ANNs). However, for many data applications the important features are of a more local character (i.e. a given pixel in an image is most likely correlated with neighboring pixels), and the concept of Convolutional Neural Networks (CNNs) has been introduced (Goodfellow et al., 2016). The core of a CNN is a hierarchy of local filters being trained to extract the essential features of the training data relevant for the application in

question. Such networks have recently attracted a great deal of attention in various fields of science and engineering, including geophysics. Many successful applications have been reported due to easy access to user-friendly open-source software, such as Google Tensorflow (Abadi et al, 2016) and PyTorch (Paszke et al., 2017), in combination with increasingly powerful hardware (CPU and GPU) available at fairly moderate cost.

Within the field of seismic image classification and interpretation, CNNs have already proved to be useful. Qian et al. (2018) proposed the use of a deep convolutional autoencoder (DCAE) network for seismic facies recognition based on prestack seismic data. Waldeland et al. (2018) demonstrated how CNNs could be used to classify different seismic textures with special emphasize on salt bodies. Xiong et al. (2018) trained a CNN to automatically detect and map fault zones using 3D seismic images, whereas Wu et al. (2019) proposed to use CNNs to pick the first arrivals of microseismic events. Baardman et al. (2018, 2019) proposed the use of a CNN to classify data patches in a "blended" and "non-blended" class. A second, regression based, CNN was then employed to deblend the "blended" patches, but only synthetic data were considered.

Although not yet fully investigated, the use of CNNs in seismic noise attenuation has also started to develop. Liu et al. (2018) used a 3D CNN architecture to remove random noise from a 3D poststack seismic data set. Ma et al. (2018) managed to attenuate multiples, linear noise and random noise simultaneously through the use of a CNN. However, only controlled data were employed and both the training and test data sets were computed from the same model. Slang et al. (2019) employed marine seismic field data and demonstrated successful applications of deblending and denoising using CNNs. Within the area of seismic data

interpolation and reconstruction, Mandelli et al. (2018, 2019) proposed to reconstruct missing seismic traces in the prestack domain by employing a convolutional autoencoder. They applied this network to solve the joint problem of synthetic data interpolation and Gaussian-noise attenuation. Wang et al. (2018 a, b) introduced an eight-layer residual learning network (ResNet) based on CNNs to interpolate seismic data without aliasing.

From this review of the seismic CNN literature, it follows that the use of field data is rather limited. Moreover, applications within denoising are dominated by the removal of Gaussian type noise. Such noise is of limited interest in real seismic data applications where we are normally faced with various types of coherent or semi-coherent noise. Thus, the performance of a CNN on more realistic seismic denoising tasks needs to be more properly addressed. In many of the current studies published, the amount of data used is not representative for problems in the seismic field. A fundamental requirement in ML is the access to a statistically sufficient set of data that can be split into feasible subsets for training, validation and testing purposes. Otherwise, there is a high chance of overfitting.

In this paper, the feasibility of employing CNNs within the area of deblending is investigated. To fully ensure that this study is as realistic as possible, only real marine field data is used. The data diversity is also properly addressed by using 21000, 4500 and 1500 images respectively for training, validation and testing.

This paper is organized as follows. In the first section a brief description of the main CNN concepts is given, followed by a section describing and discussing the actual architecture being employed to deblend seismic data. As already discussed, resorting data to obtain incoherency in the blended contribution, transforms the deblending problem to one of removing

incoherent noise. In this paper, sorting to the common channel domain is performed before the actual training by the CNN.

In the third section examples of employing the proposed network to blended field data with time delays of 1.8s ± 0.2s random jitter are presented. The effect of SNR on the deblending quality is also discussed. In the fourth section we illustrate the robustness of the proposed approach. The same trained network is then applied to a new blended field data set from a different geological area and with slightly different time delays of 2.0s ± 0.25s random jitter. The results obtained are comparable in deblending accuracy. The fifth section compares the CNN approach to deblending with the results obtained employing conventional denoising algorithms. Finally, a set of conclusions is given.

## BASIC CONCEPTS OF CNN

Fully connected layers are frequently used in deep learning ANNs, but do not represent an ideal architecture for seismic data for two main reasons. Firstly, a fully connected layer is computationally significantly more demanding than a convolutional layer, since each neuron is connected to every neuron in the previous layer and each connection has its own learnable parameter, commonly referred to as its weight. By contrast, each neuron in a convolutional layer is only connected to a few neurons in the previous layer, and shares the same set of weights (cf. Figure 1). Seismic data sets tend to be large where each shot gather may typically contain more than $10^6$ data samples, making the use of fully connected layers very challenging given the large memory requirements and need for high-performance computing.

Secondly, as suggested by the name, a convolutional layer applies a convolution operation on the input based on a bank of filter kernels (also called convolution matrix or mask).

Let $A \in ¡^{3\times3}$ denote the input image with elements $a_{k,l}$ for $k,l \in \{0,2\}$, $O \in ¡^{2\times2}$ denote the output image with elements $o_{m,n}$ for $m,n \in \{0,1\}$ and $W \in ¡^{2\times2}$ denote the filter kernel with weights $w_{i,j}$ for $i,j \in \{0,1\}$. Figure 1 gives an example of how the 2 × 2 filter works on the 3 × 3 image with stride equal to 1, to give the output image (with mirrored kernel to ensure convolution). The stride is defined by the distance between two consecutive positions of the filter kernel (Dumoulin and Visin, 2018). The 2D convolution operation (Figure 1a) can be represented by the neural network configuration (Figure 1b) where the filter weights are represented by color coding. We can take the orange square inside the red box as an example. In this case, the result from the convolution comes from the linear combination $w_{1,1}a_{1,1} + w_{1,0}a_{1,2} + w_{0,1}a_{2,1} + w_{0,0}a_{2,2} = o_{1,1}$, where the kernel is linked to the orange neuron inside the red circle in the network by the gray arrow. As shown in Figure 1b, only four blue neurons in the previous layer are connected through weights (filter coefficients) with the orange neuron inside the red circle (with weights being color coded according to the color scheme chosen for the filter coefficients).

According to Goodfellow et al. (2016), this type of architecture makes CNNs well suited for 2D images where neighboring pixels are connected to form local patterns. It should therefore be possible to deblend seismic data after common channel resorting, where the unblended data (data from the first source) exhibits a continuous and coherent form but the blending noise (data from other sources) manifests itself as incoherent contributions.

The term convolutional neural network (CNN) is used in a broad sense. In fact, all artificial neural networks containing one or more convolutional layers can be classified as CNNs. A feedforward neural network consists of basic units represented by the neurons that

are stacked into layers, with the output of one layer serving as the input for the next one. The complete neural network can be thought of as a complicated nonlinear transformation of the input into a predicted output that depends on the learnable weights and biases of all the neurons in the input layer, the hidden layers and output layer (Mehta et al., 2019).

Consider a training data set $\{T_i, \tilde{T}_i\}_{i=1}^{M}$ where $T$ and $\tilde{T}$ define the clean (ground truth) and contaminated data respectively. Since, in our case, the contamination in $\tilde{T}$ is spatially discontinuous in the chosen gather domain, we want to construct and train a function (network) $f_{W,b} : \tilde{T}_i \rightarrow T_i$, which preserves the spatially continuous character in the seismic image. The function $f_{W,b}$ is in our case based on a conventional feed forward CNN architecture with no dense layers. Let $N_l$ denote the number of features and $k_l = 1, 2, ..., N_l$ denote the $k$'th convolution filter in layer $l = 1, 2, ..., L$. The feature mapping from one arbitrary layer to the next can be summarized by the expression

$$Z_{k_l}^{[l]} = \sum_{k_{l-1}=1}^{N_{l-1}} \left( W_{k_l}^{[l]} * A_{k_{l-1}}^{[l-1]} \right) + B_{k_l}^{[l]}, \tag{1}$$

where $W_{k_l}^{[l]} \in \mathbb{R}^{f_{l-1} \times f_{l-1}}$ contains the weights and $B_{k_l}^{[l]}$ is a matrix of same size as $Z_{k_l}^{[l]}$ containing the biases. The notation $*$ denotes the convolutional process. In equation 1, $A_{k_{l-1}}^{[l-1]}$ represents one of the activations from the previous layer. The activation is defined by a non-linear transformation on all $k_l = 1, 2, ..., N_l$ mappings of the feature maps, and defines the output from layer $l-1$ to layer $l$. The activation of layer $l$ can be represented by the general expression

$$A_{k_l}^{[l]} = \varphi^{[l]}(Z_{k_l}^{[l]}), \tag{2}$$

where $\varphi^{[l]}$ is the non-linear function. In our case, we chose the Leaky Rectified Linear Unit or Leaky ReLU (Maas et al., 2013) defined as

$$\varphi^{[l]} = \max(Z_{k_l}^{[l]}, \alpha Z_{k_l}^{[l]}). \tag{3}$$

It is a modified version of the more conventionally used ReLU function where a slope is introduced for negative arguments. In the seismic case, we observed that the slope value $\alpha$ seems to benefit from being larger than the small values advocated in the literature for conventional images. In fact, we employed $\alpha = 0.4$, as opposed to the typical value of $\alpha = 0.01$ used in the non-seismic case. The use of the conventional ReLU activation function may cause a problem called "dead neurons". In the case where input to a ReLU with its weights is negative, the output will be 0, causing the gradient also to be 0. If instead Leaky ReLU is used, the gradient will never be 0 and this problem is avoided.

The convolution process of going from one arbitrary layer $l-1$ to layer $l$ represented by the first term on the right-hand-side of equation 1 is illustrated in Figure 2 for one single element in the matrix $Z_{k_l}^{[l]}$. The matrix $W_{k_l}^{[l]}$ represents the $k_l$'th filter kernel spanning all activations in layer $l-1$ producing each of the $k_l = 1,2,...,N_l$ feature maps in layer $l$.

The logistic function, also known as the sigmoid function, is also a common activation function in neural networks. The sigmoid activation function is defined as

$$\varphi^{[l]} = \sigma^{[l]} = \frac{1}{1+e^{-Z_{k_l}^{[l]}}}. \tag{4}$$

In our case the network performance improved when the sigmoid function was used in the output layer, which defines the predicted clean image

$$\mathbf{\mu} = \sigma\left[\sum_{k_{L-1}=1}^{N_{L-1}} \left(W_{k_L}^{[L]} * A_{k_{L-1}}^{[L-1]}\right) + B_{k_L}^{[L]}\right]. \tag{5}$$

We fit the predicted data by minimizing the $L_2$ loss of the difference between the clean target images $T$ and the prediction $\mathbf{\mu}$,

$$L_2(W,B) = \frac{1}{2}\left\| T - \hat{T} \right\|_2^2 . \quad (6)$$

In order to find the weights and biases that minimize equation 6, we train the network using a first-order gradient method for stochastic optimization, known as RMSprop (Tieleman and Hinton, 2012).

For the training process we split the training data into three data sets: a training set, a validation set and a test set. The data sets are constructed by sorting the training data randomly, and distributing them such that approximately 80% of the total number of data samples is used for training, 15% for validation and 5% for testing. While training, the network adjusts the weights and biases based on the difference between ground truth and the predicted output. The final model (in terms of weights and biases) is chosen according to the best fit on the validation set. After the training and validation phases are complete, the performance of the network is checked by applying the model to the independent test data.

When discussing a CNN architecture, it is important to notice that seismic data contains very different structural information compared to conventional images. To illustrate these differences, Figure 3a and Figure 3b show a conventional image and its seismic counterpart. On direct comparison, the seismic image contains a much narrower band in both temporal and spatial frequencies making the texture different from the conventional image. Equally important, the conventional image is in colors (RBG) so the input to a CNN network will be three channels, one for each color, opposed to the seismic case which is only represented by one channel of grey-scale values. Also, the dynamic range of a seismic image is often large compared to that of a picture, especially for prestack data, where amplitudes within a typical gather vary by three orders of magnitude or more. This means that the blending noise we try to

remove might typically be 1000 times stronger when compared to the unblended signals underneath. Deblending is therefore a non-trivial signal processing problem that, as mentioned in the introduction, is receiving a lot of attention from both industry and academic research. These major differences imply that well-established CNN architectures employed within image processing may not be ideal when applied to seismic gathers. Thus, new designs need to be developed and tested for each application in question. In the next section, a CNN architecture developed for the deblending problem is introduced and discussed.

## A CNN ARCHITECTURE FOR SEISMIC DEBLENDING

We now present and discuss the proposed CNN architecture for denoising/deblending of seismic data. The input data are assumed to be resorted from the common source to the common channel domain to make the blending noise incoherent. To reduce the size of the seismic data volume, the data were resampled from 2ms to 4ms. Moreover, the data were also segmented into smaller subsets of size 800 time samples × 40 traces, and amplitudes were normalized to fall in the range 0 to 1 using the following equations,

$$blended_{norm} = \left( \frac{blended}{maxxer} + 1 \right) / 2 \ , \quad blended_{norm} \in [0,1] , \tag{7}$$

$$truth_{norm} = \left( \frac{truth}{maxxer} + 1 \right) / 2 \ , \quad truth_{norm} \in [0,1] , \tag{8}$$

where $maxxer$ is the maximum absolute value of the blended data and ground truth/unblended data. We use $blended_{norm}$ and $truth_{norm}$ as the input of the network, and this process is reversible. The output of the network can be denormalized by

$$output_{denorm} = \left( output \times 2 - 1 \right) \times maxxer . \tag{9}$$

The complete CNN design employed in this study is shown schematically in Figure 4.

The proposed CNN has eight convolutional layers in total and can be classified as a deep network. The convolutional operations to produce the first five hidden layers have 64 kernels with size 3 × 3, which for the last two hidden layers are reduced to 32 kernels with size 3 × 3. The Leaky Rectified Linear Unit (Leaky ReLU) is used as an activation function in every convolutional layer except the last one where the Sigmoid function is employed. In addition, Batch Normalization (BatchNorm) is used in the initial part of the network. BatchNorm is a widely adopted technique that enables faster and more stable training of deep neural networks (DNNs). According to Ioffe and Szegedy (2015), BatchNorm addresses the problem called internal covariate shift by normalizing layer inputs. In traditional deep networks, high learning rates may result in gradients that explode or vanish, as well as solutions stuck in local minima. BatchNorm helps to avoid zero values in the network, which easily appear due to the large dynamic range in prestack (blended) data gathers. A disagreement exists in the literature on whether covariate shift enables BatchNorm to improve training. Santurkar et al. (2018) suggested that BatchNorm makes the optimization landscape significantly smoother, thus inducing a more predictive and stable behavior of the gradients, allowing for faster training.

Compared to typical CNNs employed in conventional image analysis, no downscaling is applied in our model. Take max pooling (Boureau et al., 2010; Scherer et al., 2010) as an example. The objective of adding max pooling to image classification models is to down-sample an input representation, reducing its dimensionality, and also to help avoid overfitting by providing an abstracted form of the representation. However, in seismic deblending, it is important to preserve as much as possible of the geological information while removing blending noise. Thus, in our case, the network is designed without downscaling to reduce

potential blurring and precision loss. The initial learning rate was set at 0.001 and automatically multiplied by a factor 0.9 every second epoch.

Having introduced the key components of the CNN used, we give a more detailed discussion of the various main design choices.

**Filter size**

Different filter sizes were tested as part of the design process for the CNN. Larger filter sizes of 7 x 7 and 5 x 5 were tested but the performance of the trained network was found to be poorer than when 3 x 3 filters were used. Figure 5 shows an example of a deblended source gather (after resorting from common channel gathers) for three different sizes of the convolutional filters.

**Number of filters**

Different combinations of filter banks were tested in order to optimize the design. The performance was quantified using the loss. Figure 6 shows an example of the loss function (training and validation) for two sets of filter combinations, respectively denoted as model 1 and model 2: 64 and 32 (the one used in the actual CNN) and 32 and 16.

**Number of layers**

It is not guaranteed that adding more layers or more neurons to a CNN will improve its performance. Redundancy will result in increased training time and the waste of computational power. In order to analyze the effects of adding more layers to the network, we carried out a comprehensive quality control of the feature maps output from each layer. The main idea was to obtain a maximum of complementary features and avoid 'dead' convolutional filters (e.g. with no action on the data). Figure 7 shows an example of a collection of feature maps for the

last hidden layer in the final design of our CNN. Because of limited space, only six out of 32 panels are shown here. We can see how the network decomposes the data partially in character and partially in frequency bands, with some features enhancing the blended noise whereas others enhance the signal to be recovered.

## FIELD DATA EXAMPLES

In the first part of this study we used 1300 unblended marine split-spread shot gathers from a 2017 survey in the Barents Sea (Vinje et al., 2017). From this set of data, we constructed blended shots by adding two consecutive shots with a fixed delay time perturbed with a predefined random jitter. By changing the delay time, various source configurations could be simulated. In the example considered, the delay time was set to 1.8s ± 0.2s of random jitter. This implies a rather challenging case, where the blended contribution appears at larger traveltimes, thus superimposing the weaker reflections in the unblended source gathers (i.e. the events to be recovered). The training, validation and test data sets consisted of 21000, 4500 and 1500 images respectively with 40 traces per image after sorting to the common channel domain. In fact, we can choose any number of traces per image, but too few traces will be insufficient to capture local geology and too many traces will be difficult to fit into computer memory. The training process (35 epochs) employing a deep CNN requires significant computational power, and for this particular test the run time was approximately 140 hours on a standard CPU (Intel Xeon E5-1620 0, 3.60Hz, 10 MB Cache, 4 Cores, 8 Threads), but only 7 hours on a fairly modern GPU (Nvidia GeForce GTX TITAN Black, 6GB). However, once the network was properly trained, deblending of a single gather could be done in nearly real time.

As already discussed, a more optimized denoising problem is achieved by resorting the data into the common channel domain. In this domain, the blending noise will transform from coherent to incoherent events.

Figure 8 shows an example of a result obtained using the trained network in the common channel domain. Figures 8a-c represent the ground truth (i.e. the unblended source gather), the blended source gather and the deblended result output from the CNN, respectively. Moreover, Figures 8d and 8e show respectively a difference plot between the ground truth and the deblended result, and the noise removed by the proposed CNN. As part of this combined figure, the amplitude spectra of the ground truth and the deblended result are also shown. We can observe that the network has performed well overall. To further quantify the quality of the deblending, we calculate the error via the following equation,

$$error = \left( \frac{100 \times value_{diff}}{value_{standard}} \right) \%, \tag{10}$$

where $value_{standard}$ is the sum of the absolute amplitude values of ground truth and $value_{diff}$ is the sum of the absolute amplitude values of the difference between the output and ground truth. In case of the example shown in Figure 8, we calculated the error to be 0.226% which is a quite encouraging result.

The actual noise removed by the network is shown in Figure 8e. Its incoherent nature can easily be seen from this figure. The corresponding amplitude spectrum is also shown below. Direct comparison with the spectra computed from the ground truth and the removed noise show strong correlation. Thus, as expected, this type of noise is far from the ideal Gaussian distribution.

To further investigate the performance of our CNN, we apply the trained network on

an ensemble of common channel test data and resort back to the shot domain. Figure 9 shows an example of such a deblended source gather. The sequence of subfigures is the same as in Figure 8. It can be seen that the blending noise has been mostly removed. In this example the error was calculated to be 0.241%. However, weak residuals are still present, and we observe a pattern of blank stripes in the deblended shot gather, which coincide with the very strong water-bottom reflections from the N+1 shot.

To add some more insight into the observations made in Figure 9, we repeated the experiment but with the blended contribution being scaled down by a given factor before being superimposed on the ground truth. Four different scenarios were investigated with the value of the factor being 0.8, 0.6, 0.4 and 0.2 respectively. As the factor decreases, the SNR of the blended data increases (the events of the ground truth become stronger relative to the blended events).

Figure 10 shows a summary of the results obtained after resorting to the shot domain. To enhance the comparison, only a zoomed part of the image within the target zone (red box in Figure 9) is shown for each experiment. In order to have a measure of the deblending accuracy, we calculated again the error as defined in equation 10. In case of Figures 10a-d this measure took the values 0.043%, 0.015%, 0.012% and 0.007% respectively. It can clearly be seen, that the SNR is the major factor controlling the quality of the final result.

Before closing this data example section, the important issue of robustness needs to be addressed. Two questions can be raised. Firstly, how well will a trained network perform on data from another survey and different geological area? Secondly, how well will a trained network perform on another blending case where the blended shots have blending noise in the

top part of the data? If we can avoid time-consuming retraining it will make the use of a CNN much more attractive, since the data processing time will be dramatically reduced compared to a conventional approach.

To investigate how robust and adaptive our proposed network is, we firstly used the trained model from the Barents Sea study discussed above to deblend data from a different survey campaign. This new data set had a slightly different delay time of 2.0s ± 0.25s random jitter, but more importantly was acquired in the North Sea (Dehlie et al., 2018). Thus, the geology of this latter area is very different from that of the Barents Sea, being separated by a distance of almost 2000km. Figure 11 shows an example of a deblended result in the common channel domain. Correspondingly, Figure 12 shows a deblended shot gather after resorting back to the shot domain. Direct comparisons between Figures 8 and 11 and between Figures 9 and 12 show that the new deblended data are of similar quality.

Our second test of robustness was to deblend shots which had blending noise in the top part of the record. For this, the same CNN as before was employed without retraining. Appendix A gives an example of a deblended second shot (N+1) as well as results obtained in the common channel domain. It can be seen that the network performs well but, as expected, not as well as on the first type of blended shots (N) which were originally used to train the network.

These results are encouraging and demonstrate that construction of a robust network design is feasible. The trained network has therefore learned the difference in morphological characteristics between blending noise and ground-truth signals. The key to this success has been the sorting of data to the common channel domain where the blending noise always

distributes randomly while target signals preserve their coherent nature.

## COMPARISON BETWEEN THE CNN AND COMMERCIAL ALGORITHMS

To efficiently handling the problem of deblending, the industry typically applies a combination of different processing algorithms. This implies a computer-intensive approach to solving this problem. In order to further test the quality of our proposed CNN in a fair way, we compare its performance with the results obtained employing two industry denoising algorithms, since the N+1 shots appear as incoherent noise in the common channel domain. The first technique considered was based on F-X prediction filtering (Gulunay, 1986) and the second method was based on the concept of projective filtering in the F-X domain (Traonmilin and Gulunay, 2013).

Figure 13 shows the ground truth, blended data, blending noise and the deblended results obtained employing the proposed CNN in the common channel domain. In addition, the results obtained using the two industry approaches mentioned above are shown. To quantify the difference between the output and ground truth, we calculate the error via equation 10 as in the previous section. In case of the CNN result (Figure 13e) the error is only 0.226%, and less than the corresponding errors in case of F-X prediction filtering (Figure 13h) (0.366%) and projective filtering result (Figure 13k) (0.272%). In Table 1, a comparison of the denoising accuracies between the proposed CNN, F-X prediction filtering and projective filtering is given for typical shot gathers. Figure 13i shows that F-X prediction filtering removed noise that was 1.459 times stronger than the actual blending noise level, thus too much. From direct comparison between Figures 13c and 13l, it is obvious that projective filtering only removed part of the blending noise. Finally, we can observe that the CNN shows relatively better

deblending result compared to both industry methods. However, the performance of the CNN is not perfect. Although this approach is able to suppress the noise quite well, some coherent energy is lost particularly in the shallower parts.

Besides processing accuracy, computational time is also an important parameter when evaluating an algorithm. We have already seen that our CNN, if trained on data from one geological area, may work well on field data from a survey acquired across a very different geological setting. Thus, as soon as the training phase is completed, the actual application of the CNN approach has the potential to be regarded as a real-time application for a given image. In Table 2, a summary comparing the corresponding computational times can be found. From this latter table it follows that our CNN can process data more than 300 times faster than the two conventional algorithms once it has been trained.

## CONCLUSION

In this study we have investigated the idea of employing a CNN to solve the problem of deblending seismic data. Straightforward use of networks designed for conventional image processing will not be optimal. We have therefore designed our own network taking into account the special characteristics of seismic data.

The proposed CNN architecture was trained on numerically blended field data, and then its performance was verified on a set of test data. Because of the powerful computational ability of modern GPUs, the complete training of our network took only seven hours. Minor and weak residuals were observed in the data when employing large delay times between blended shots. This could be explained by considering the SNR between the ground-truth signals and the superimposed deblending noise.

To further investigate our network, we compared our results with the results obtained by using two conventional industry denoising algorithms. We then observed that the proposed CNN performed better when it comes to deblending accuracy and also demonstrated a favorable computational time after being properly trained. The fast-computation makes the proposed CNN suitable for fast-track processing and onboard/field deblending which is not always possible today. We also demonstrated that our network is robust. While it was trained on data acquired from the Barents Sea with delay times around 1.8s, application on test data acquired from the North Sea with delay times around 2.0s still gave very reasonable and encouraging results. In addition, we used the same trained network to deblend shots with blending noise in the top part of the data, which again demonstrated the robustness of our proposed CNN design.

Deblending of seismic data requires high accuracy, and any significant error in geological information will damage the quality of the final seismic image. Building a robust network architecture, organizing high-quality training data, and applying appropriate preprocessing are all essential for a successful learning process. The ground truth without any noise should ideally be known. This is difficult to achieve when we work with real data, which will inevitably contain some noise contamination.

## ACKNOWLEDGEMENTS

The authors thank CGG Data Library for permission to use the data shown in this article. The CNNs studied in this paper were implemented by using Keras developed by Google.

## APPENDIX A

## DEBLENDING RESULTS OF THE N+1 SHOTS

In this appendix we consider the second blended (N+1) shot that had blending noise in the top part of the data. The same trained CNN as before was employed to deblend such data without retraining. Figure A-1 shows the deblending results obtained in the common channel domain and Figure A-2 shows corresponding results after resorting to shot domain. It can be seen that the network performs well but, as expected, not as well as on the first type of blended shots (N) which were originally used to train the network.

## REFERENCES

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, 2016, Tensorflow: A system for large-scale machine learning: 12th Symposium on Operating Systems Design and Implementation, USENIX, Proceedings, 265-283.

Akerberg, P., G. Hampson, J. Rickett, H. Martin, and J. Cole, 2008, Simultaneous source separation by sparse Radon transform: 78th Annual International Meeting, SEG, Expanded Abstracts, 2801-2805.

Andersson, F., K. Eggenberger, D. J. Manen, and J. O. A. Robertsson, 2016, Seismic apparition dealiasing using directionality regularization: 86th Annual International Meeting, SEG, Expanded Abstracts, 56-60.

Baardman, R., 2018, Classification and Suppression of Blending Noise Using CNN: First EAGE/PESGB Workshop Machine Learning, EAGE, Extended Abstracts, M11.

Baardman, R., and C. Tsingas, 2019, Classification and Suppression of Blending Noise Using Convolutional Neural Networks: Middle East Oil and Gas Show and Conference, SPE, Conference Papers, 1-8.

Barbier, M. G., 1982, Pulse coding in seismology: Springer.

Beasley, C. J., 2008, A new look at marine simultaneous sources: The Leading Edge, 27, no. 7, 914-917.

Berkhout, A. J., 2008, Changing the mindset in seismic data acquisition: The Leading Edge, 27, 924-938.

Berkhout, A. J., G. Blacquière, and D. J. Verschuur, 2010, Multi-scattering illumination in blended acquisition design: 80th Annual International Meeting, SEG, Expanded Abstracts, 1251-1255.

Boureau, Y. L., J. Ponce, and Y. LeCun, 2010, A theoretical analysis of feature pooling in visual recognition: 27th International Conference on Machine Learning, ICML, Proceedings, 111-118.

Candes, E., L. Demanet, D. Donoho, and L. Ying, 2006, Fast discrete Curvelet transforms: Multiscale Modeling and Simulation, 5, 861-899.

Chakraborty, A., and D. Okaya, 1995, Frequency-time decomposition of seismic data using wavelet-based methods: Geophysics, 60, no. 6, 1906-1916.

Chen, Y., S. Fomel, J. Hu, 2014, Iterative deblending of simultaneous-source seismic data using seislet-domain shaping regularization: Geophysics, 79, no. 5, V179-V189.

Chen, Y., 2015, Deblending using a space-varying median filter: Exploration Geophysics, 46, no. 4, 332-341.

Dehlie, P. E., V. Danielsen, H. Shen, and T. Elboth, 2018, Hexasource: Wide tow-dithered six-source marine acquisition in the Barents Sea: Technical Program, SEG, Expanded Abstracts, 46-50.

Doulgeris, P., and K. Bube, 2012, Analysis of a coherency-constrained inversion for the separateion of blended data: discovering the leakage subspace: 74th Annual International Conference and Exhibition, EAGE, Extended Abstracts, C14.

Dumoulin, V., and F. Visin, 2018, A guide to convolution arithmetic for deep learning: arXiv, 1603.07285.

Fomel, S., and Y. Liu, 2010, Seislet transform and seislet frame: Geophysics, 75, no. 3, V25-V38.

Goodfellow, I., Y. Bengio, and A. Courville, 2016, Deep learning: MIT press.

Gulunay, N., 1986, FXDECON and complex Wiener prediction filter: 56th Annual International Meeting, SEG, Expanded Abstracts, 279-281.

Gwardys, G., 2016, Convolutional Neural Networks backpropagation: from intuition to derivation, https://grzegorzgwardys.wordpress.com/, accessed 19 June 2016.

Helgason, S., 1999, The Radon transform: Springer.

Herrmann, F. J., Y. A. Erlangga, and T. T. Lin, 2009, Compressive simultaneous full-waveform simulation: Geophysics, 74, no. 4, A35-A40.

Huo, S., Y. Luo, and P. G. Kelamis, 2012, Simultaneous sources separation via multidirectional vector-median filtering: Geophysics, 77, no. 4, V123-V131.

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

Kutyniok, G., and W. Q. Lim, 2011, Compactly supported shearlets are optimally sparse: Journal of Approximation Theory, 163, no. 11, 1564-1589.

LeCun, Y., Y. Bengio, and G. Hinton, 2015, Deep learning: Nature, 521, 436-444.

Liu, D., W. Wang, W. Chen, X. Wang, Y. Zhou, and Z. Shi, 2018, Random noise suppression in seismic data: What can deep learning do?: 88th Annual International Meeting, SEG, Expanded Abstracts, 2016-2020.

Ma, J., 2018, Deep Learning for Attenuating Random and Coherence Noise Simultaneously: 80th Annual International Conference and Exhibition, EAGE, Extended Abstracts, Tu P7, 07.

Maas, A. L., A. Y. Hannun, and A. Y. Ng, 2013, Rectifier nonlinearities improve neural network acoustic model: Proceedings of International Conference on Machine Learning, 30, no. 1, 3.

Mahdad, A., P. Doulgeris, and G. Blacquiere, 2011, Separation of blended data by iterative estimation and subtraction of blending interference noise: Geophysics, 76, no. 3, Q9-Q17.

Mahdad, A., P. Doulgeris, and G. Blacquière, 2012, Iterative method for the separation of blended seismic data: discussion on the algorithmic aspects: Geophysical Prospecting, 60, no. 4, 782-801.

Mandelli, S., F. Borra, V. Lipari, P. Bestagini, A. Sarti, and S. Tubaro, 2018, Seismic data interpolation through convolutional autoencoder: 88th Annual International Meeting, SEG, Expanded Abstracts, 4101-4105.

Mandelli, S., V. Lipari, P. Bestagini, and S. Tubaro, 2019, Interpolation and Denoising of Seismic Data using Convolutional Neural Networks: arXiv, 1901.07927.

Maraschini, M., R. Dyer, K. Stevens, and D. Bird, 2012, Source separation by iterative rank reduction-theory and applications: 74th Annual International Conference and Exhibition,

EAGE, Extended Abstracts, A044.

Mehta, P., M. Bukov, C. H. Wang, A. G. R. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, 2019, A high-bias, low-variance introduction to machine learning for physicists: Physics Reports.

Monk, D., 2002, Lena: A seismic model: The Leading Edge, 21, 438-444.

Moore, I., B. Dragoset, T. Ommundsen, D. Wilson, C. Ward, and D. Eke, 2008, Simultaneous source separation using dithered sources: 78th Annual International Meeting, SEG, Expanded Abstracts, 2806-2810.

Neelamani, R., C. E. Krohn, J. R. Krebs, M. Deffenbaugh, J.E. Anderson, E. Mobil, and J. K. Romberg, 2008, Efficient seismic forward modeling using simultaneous random sources and sparsity: 78th Annual International Meeting, SEG, Expanded Abstracts, 2107-2111.

Neelamani, R., C. E. Krohn, J. R. Krebs, J. K. Romberg, M. Deffenbaugh, and J. E. Anderson, 2010, Efficient seismic forward modeling using simultaneous random sources and sparsity: Geophysics, 75, no. 6, WB15-WB27.

Paszke, A., S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, 2017, Automatic differentiation in PyTorch: 2017 Workshop Autodiff, NIPS, Conference Papers.

Qian, F., M. Yin, X. Y. Liu, Y. J. Wang, C. Liu, and G. M. Hu, 2018, Unsupervised seismic facies analysis via deep convolutional autoencoders: Geophysics, 83, no. 3, A39-A43.

Robertsson, J. O. A, L. Amundsen, and Å. S. Pedersen, 2016, Signal apparition for simultaneous source wavefield separation: Geophysical Journal International, 206, 1301-1305.

Santurkar, S., D. Tsipras, A. Ilyas, and A. Madry, 2018, How does batch normalization help optimization?: Advances in Neural Information Processing Systems, 31, 2483-2493.

Scherer, D., A. Müller, and S. Behnke, 2010, Evaluation of pooling operations in convolutional architectures for object recognition: International conference on artificial neural networks, Springer, Lecture Notes in Computer Science, vol 6354, 92-101.

Slang, S., J. Sun, T. Elboth, S. McDonald & L. J. Gelius, 2019, Using Convolutional Neural Networks for Denoising and Deblending of Marine Seismic Data: 81st Annual International Conference and Exhibition, EAGE, Extended Abstracts, Tu R06, 05.

Tieleman, T., and G. Hinton, 2012, Divide the gradient by a running average of its recent magnitude: COURSERA, Neural networks for machine learning, 4.2, 26-31.

Timoshin, Y. V., and A. I. Chizhik, 1982, Method for spatial seismic exploration by the multiple overlapping technique: Russia Patent SU1543357A1.

Traonmilin, Y., and N. Gulunay, 2013, Statics-preserving projection filtering: Geophysical Prospecting, 61, 346-351..

Vaage, S. T., 2005, Method and system for acquiring marine seismic data using multiple seismic sources: U.S. Patent 6,906,981.

Vinje, V., J. E. Lie, V. Danielsen, P. E. Dhelle, R. Silliqi, C. I. Nilsen, E. Hicks, and A. Camerer, 2017, Shooting over the seismic spread: First Break, 35, 97-104.

Waldeland, A. U., A. C. Jensen, L. J. Gelius, and A. H. S. Solberg, 2018, Convolutional neural networks for automated seismic interpretation: The Leading Edge, 37, no. 7, 529-537.

Wang, B. F., N. Zhang, W. K. Lu, P. Zhang, and J. H. Geng, 2018a, Seismic Data Interpolation Using Deep Learning Based Residual Networks: 80th Annual International

Conference and Exhibition, EAGE, Extended Abstracts, Th A15, 14.

Wang, B. F., N. Zhang, W. K. Lu, and J. Wang, 2018b, Deep-learning-based seismic data interpolation: A preliminary result: Geophysics, 84, no. 1, V11-V20.

Wapenaar, K., J. Neut, J. Thorbecke, 2012a, Deblending by direct inversion: Geophysics, 77, no. 3, A9-A12.

Wapenaar, K., J. Neut, and J. Thorbecke, 2012b, On the relation between seismic interferometry and the simultaneous-source method: Geophysical Prospecting, 60, no. 4, 802-823.

Wu, H., B. Zhang, F. Li, and N. Liu, 2019, Semi-automatic first arrival picking of micro-seismic events by using pixel-wise convolutional image segmentation method: Geophysics, 84, no. 3, 1-70.

Xiong, W., X. Ji, Y. Ma, Y. Wang, M. Nasher, and N. Mustafa, 2018, Seismic fault detection with convolutional neural network: Geophysics, 83, no. 5, O97-O103.

## LIST OF FIGURES

Figure 1: A schematic representation of feedforward in CNNs (adapted from Gwardys, 2016). (a) 2D convolution operation as an alternative representation of a (b) connected network. Filter weights are highlighted in four different colors.

Figure 2: Visualization outlining the convolution process of going from one arbitrary layer $l-1$ to layer $l$. The matrix $W_{k_l}^{[l]}$ represents the $k_l$'th kernel spanning all activations in layer $l-1$ producing each of the $k_l = 1,2,...,N_l$ feature maps in layer $l$.

Figure 3: Lena as a conventional color digital image (a) and its seismic counterpart (b) (Adapted from Monk, 2002).

Figure 4: A schematic of the proposed CNN for deblending seismic data. The image on the far left is the blended data input to the CNN. The image on the far right is the deblended data output. The intermediate rectangles represent convolutional layers and are annotated with key parameters. Note that BN is an abbreviation for Batch Normalization. The initial value in brackets (i.e. 64, 32 or 1) indicates the number of filters. This is followed by 3 × 3 which is the filter size. The final parameter is the activation function and is either Leaky ReLU or Sigmoid for this network.

Figure 5: Effect of varying filter size on the deblended result: (a) ground truth; (b) blended data; (c) to (e) are deblended data when 7 x 7 filters, 5 x 5 filters and 3 x 3 filters were used respectively; (f) to (h) are the differences between deblended data and ground truth when

7 x 7 filters, 5 x 5 filters and 3 x 3 filters were used respectively.

Figure 6: Training and validation loss for two different filter-bank combinations.

Figure 7: Six feature maps of (a) filter 1, (b) filter 2, (c) filter 6, (d) filter 26, (e) filter 28, (f) filter 30 out of 32 filters associated with the last hidden layer together with their corresponding frequency spectra.

Figure 8: An example of seismic deblending in the common channel domain employing CNN: (a) ground truth, (b) blended data, (c) deblended data, (d) difference between ground truth and deblended data, and (e) removed blending noise.

Figure 9: An example of seismic deblending by CNN after resorting to the shot domain: (a) ground truth, (b) blended data, (c) deblended data, and (d) difference between ground truth and deblended data. Arrow A indicates the blank stripes (low amplitude areas) and arrow B indicates the remaining blended signal.

Figure 10: (a) to (d) show deblending results when 80%, 60%, 40% and 20% blending noise is introduced, respectively. (e) shows the ground truth.

Figure 11: Deblending result of applying the trained CNN on data from the North Sea in the common channel domain: (a) ground truth, (b) blended data, (c) deblended data, (d) difference between ground truth and deblended data, and (e) removed blending noise.

Figure 12: Deblending result of applying the trained CNN on data from the North Sea after resorting to the shot domain: (a) ground truth, (b) blended data, (c) deblended data, and (d) difference between ground truth and deblended data.

Figure 13: Comparison between the CNN, F-X prediction filtering and projective filtering: (a) ground truth; (b) blended data; (c) blending noise; (d) to (f) are respectively

deblended data, the difference between ground truth and deblended data, and removed blending noise employing the CNN; (g) to (i) are respectively deblended data, the difference between ground truth and deblended data, and removed blending noise employing F-X prediction filtering; (j) to (l) are respectively deblended data, the difference between ground truth and deblended data, and removed blending noise employing projective filtering.

Figure A-1: Deblending result of applying the trained CNN on blended data that had blending noise in the top part of the data in the common channel domain: (a) ground truth, (b) blended data, (c) deblended data, (d) difference between ground truth and deblended data, and (e) removed blending noise.

Figure A-2: Deblending result of applying the trained CNN on blended data that had blending noise in the top part of the data after resorting to the shot domain: (a) ground truth, (b) blended data, (c) deblended data, and (d) difference between ground truth and deblended data.

## LIST OF TABLES

Table 1: Accuracy comparison between the CNN, F-X prediction filtering and projective filtering.

Table 2: Computational times of the CNN, F-X prediction filtering and projective filtering.

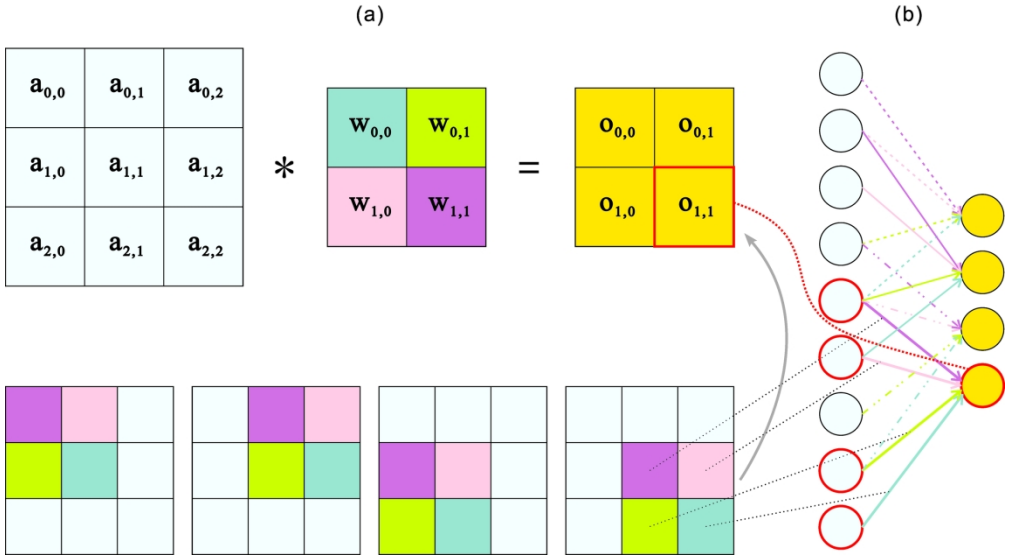


Figure 1: A schematic representation of feedforward in CNNs (adapted from Gwardys, 2016). (a) 2D convolution operation as an alternative representation of a (b) connected network. Filter weights are highlighted in four different colors.

109x60mm (600 x 600 DPI)

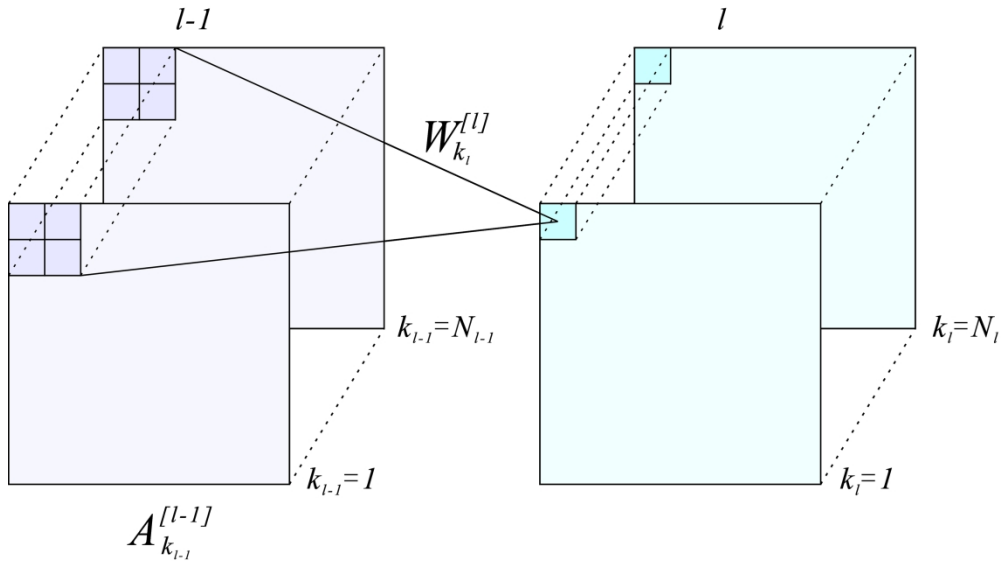


Figure 2: Visualization outlining the convolution process of going from one arbitrary layer to layer . The matrix represents the 'th kernel spanning all activations in layer producing each of the feature maps in layer .

138x78mm (600 x 600 DPI)

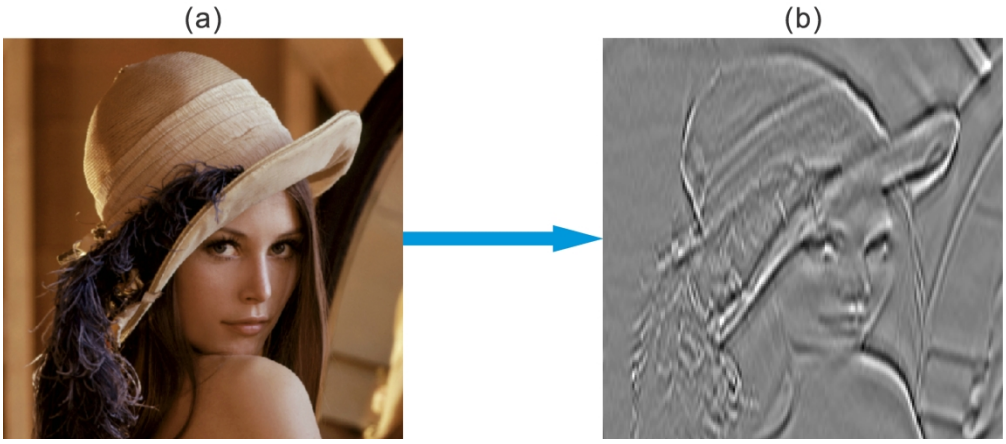


Figure 3: Lena as a conventional color digital image (a) and its seismic counterpart (b) (Adapted from Monk, 2002).

109x47mm (600 x 600 DPI)

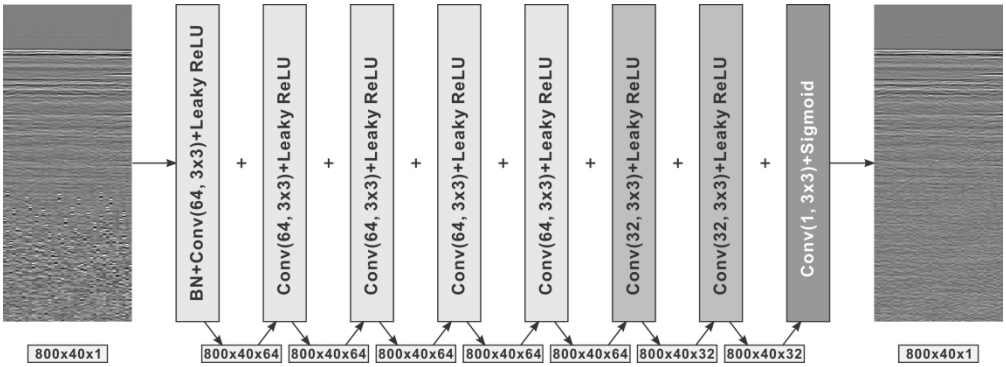


Figure 4: A schematic of the proposed CNN for deblending seismic data. The image on the far left is the blended data input to the CNN. The image on the far right is the deblended data output. The intermediate rectangles represent convolutional layers and are annotated with key parameters. Note that BN is an abbreviation for Batch Normalization. The initial value in brackets (i.e. 64, 32 or 1) indicates the number of filters. This is followed by 3 × 3 which is the filter size. The final parameter is the activation function and is either Leaky ReLU or Sigmoid for this network.

173x62mm (600 x 600 DPI)

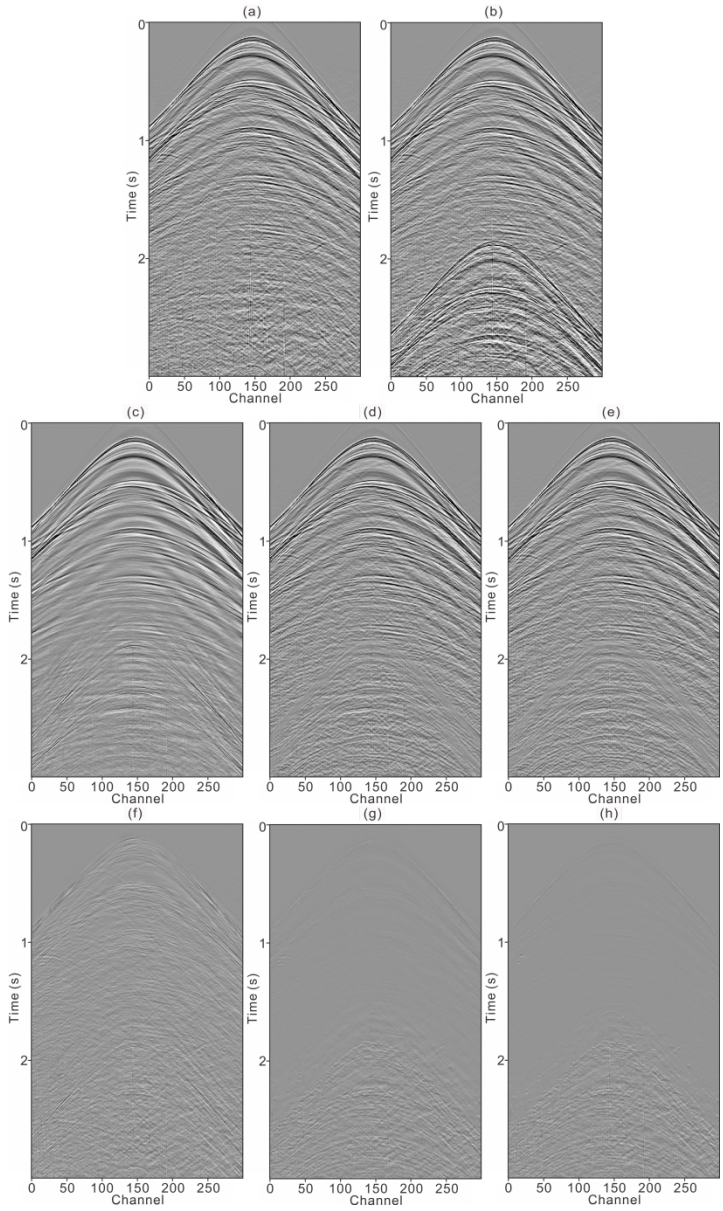


Figure 5: Effect of varying filter size on the deblended result: (a) ground truth; (b) blended data; (c) to (e) are deblended data when 7 x 7 filters, 5 x 5 filters and 3 x 3 filters were used respectively; (f) to (h) are the differences between deblended data and ground truth when 7 x 7 filters, 5 x 5 filters and 3 x 3 filters were used respectively.

155x262mm (600 x 600 DPI)

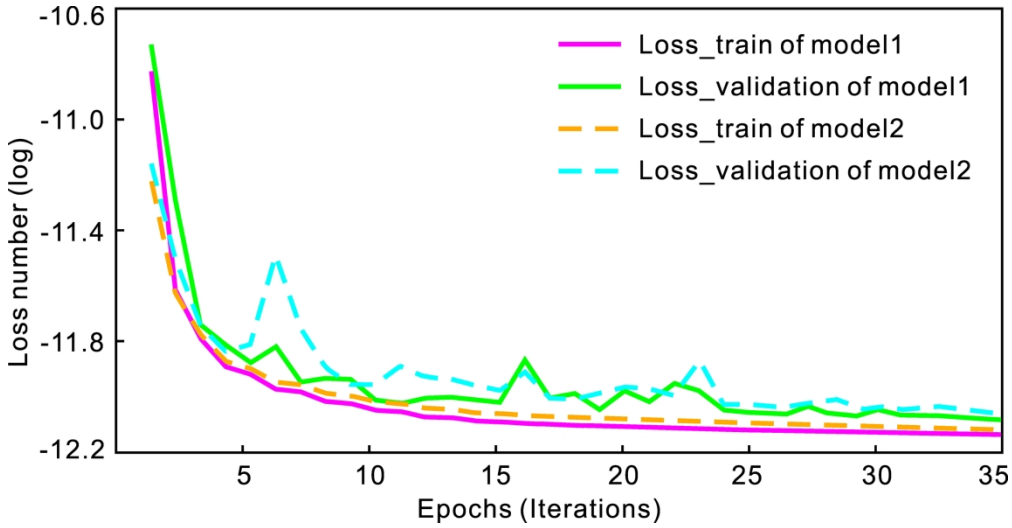


Figure 6: Training and validation loss for two different filter-bank combinations.

109x56mm (600 x 600 DPI)

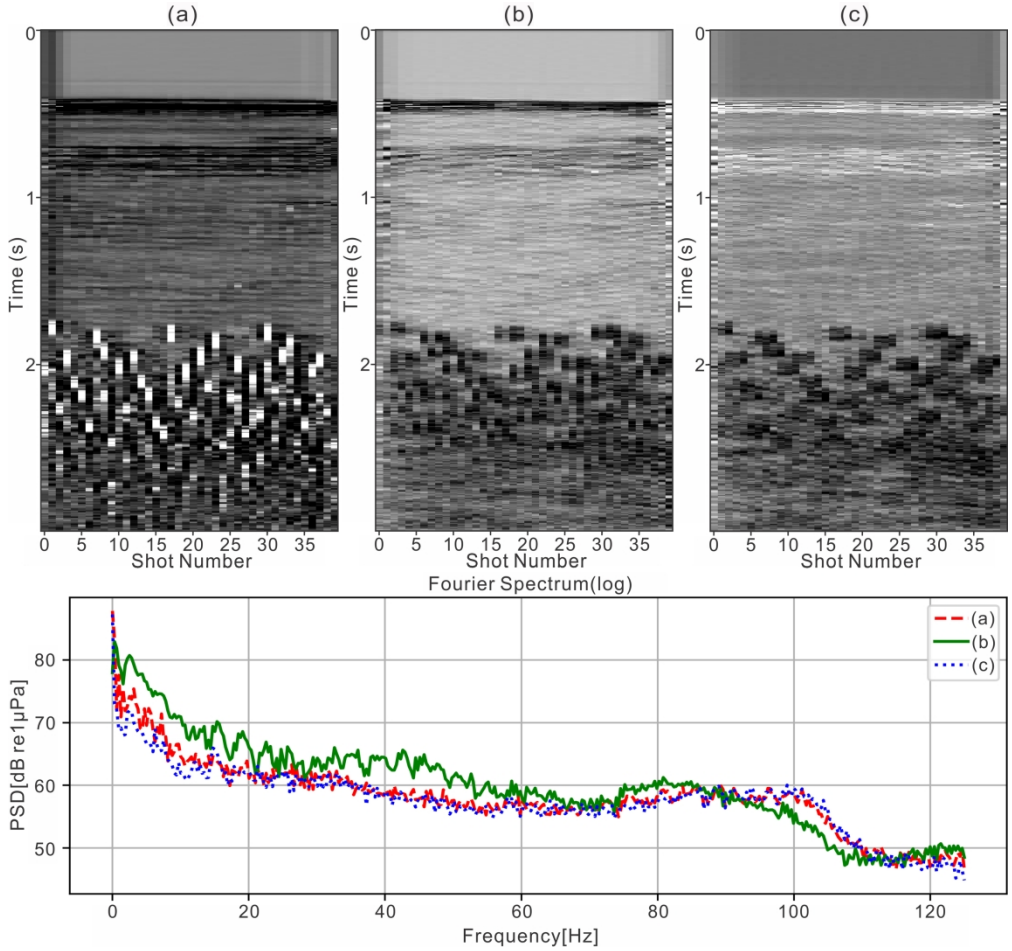


Figure 7a-c: Six feature maps of (a) filter 1, (b) filter 2, (c) filter 6, (d) filter 26, (e) filter 28, (f) filter 30 out of 32 filters associated with the last hidden layer together with their corresponding frequency spectra.

119x113mm (600 x 600 DPI)

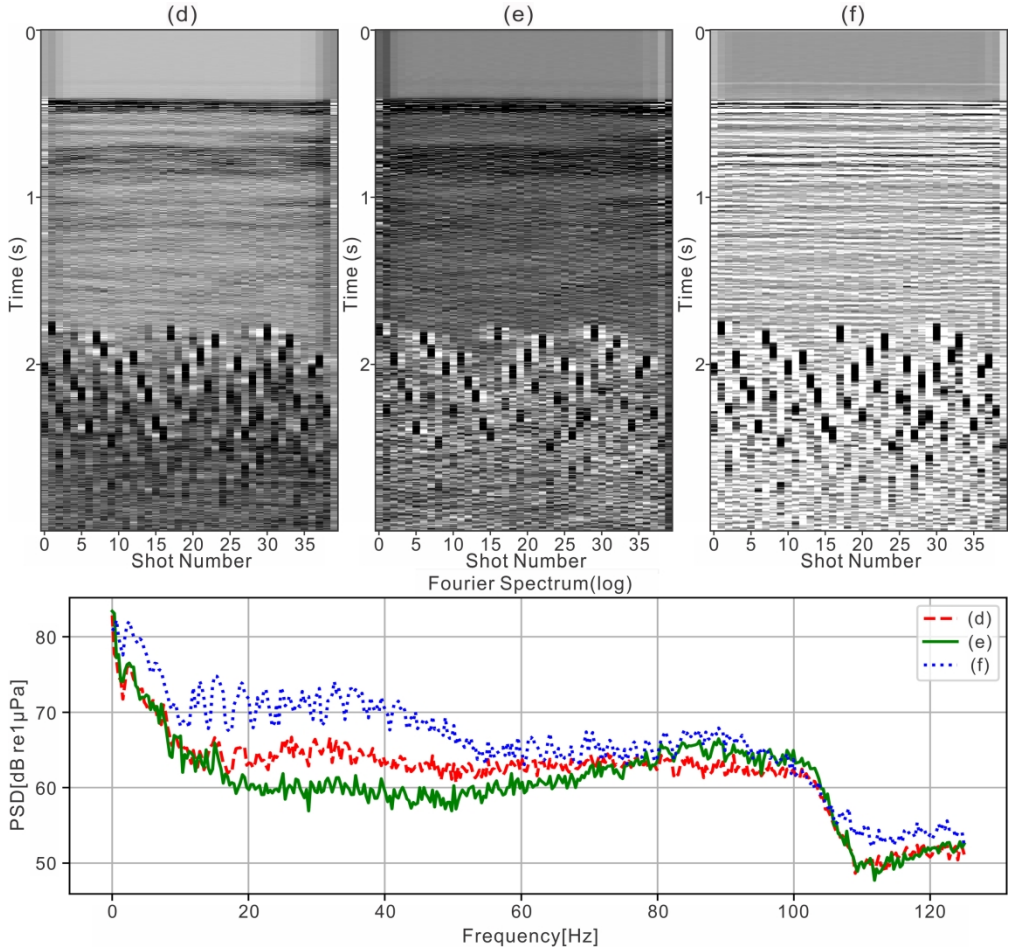


Figure 7d-f: Six feature maps of (a) filter 1, (b) filter 2, (c) filter 6, (d) filter 26, (e) filter 28, (f) filter 30 out of 32 filters associated with the last hidden layer together with their corresponding frequency spectra.

119x113mm (600 x 600 DPI)

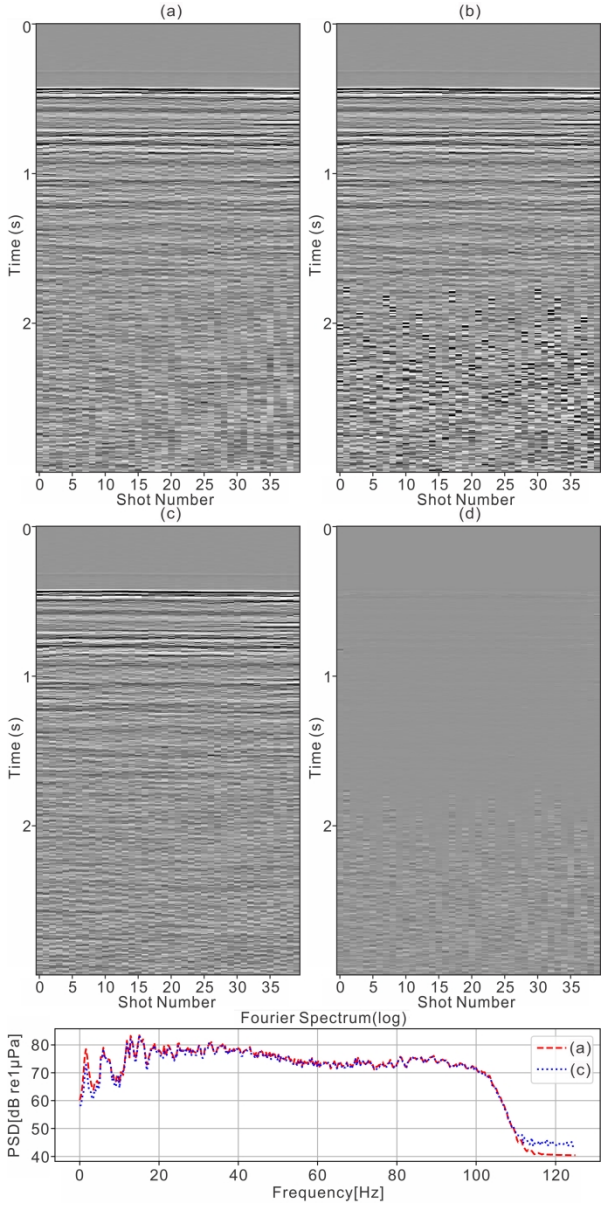


Figure 8a-d: An example of seismic deblending in the common channel domain employing CNN: (a) ground truth, (b) blended data, (c) deblended data, (d) difference between ground truth and deblended data, and (e) removed blending noise.

108x218mm (600 x 600 DPI)

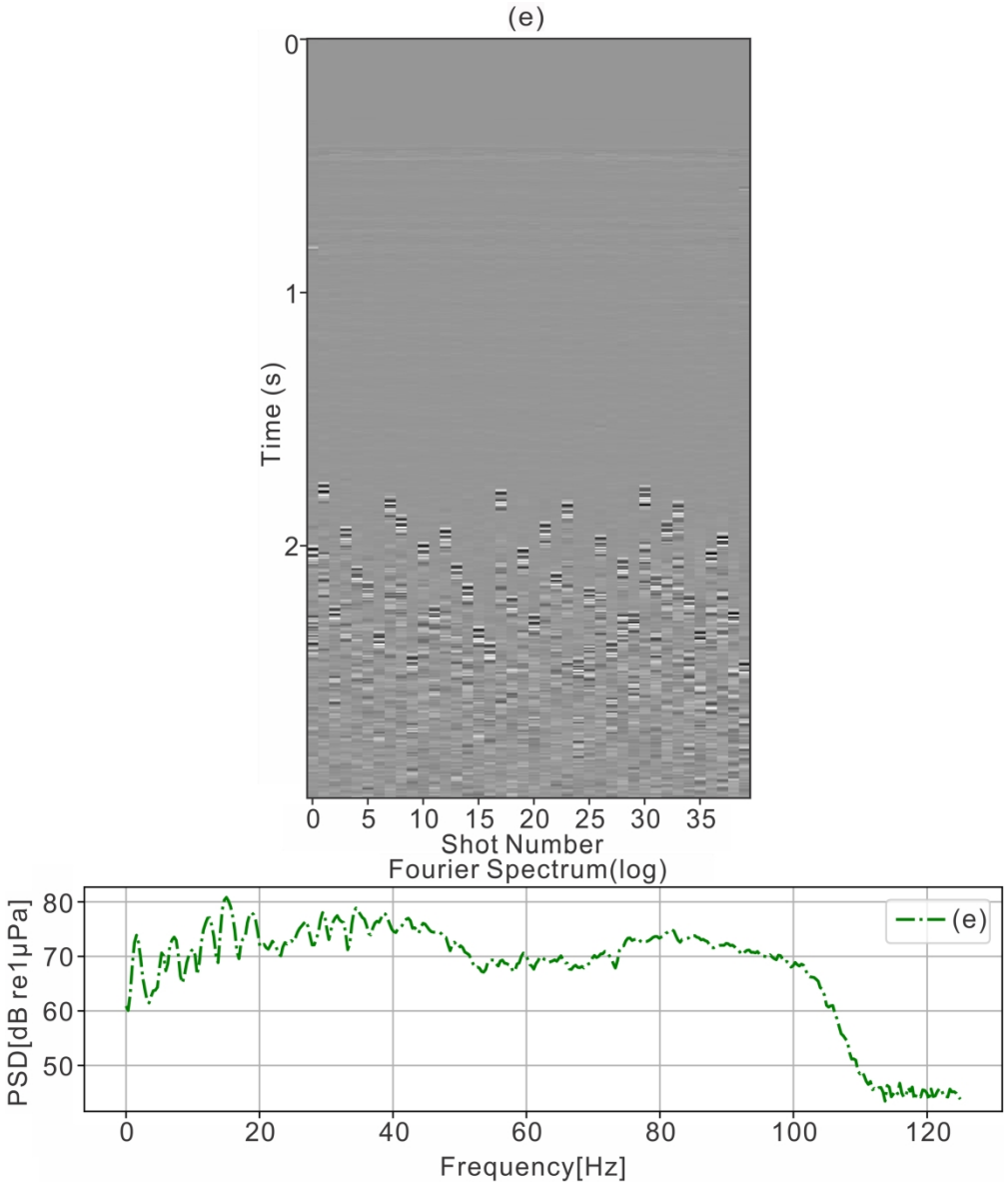


Figure 8e: An example of seismic deblending in the common channel domain employing CNN: (a) ground truth, (b) blended data, (c) deblended data, (d) difference between ground truth and deblended data, and (e) removed blending noise.

109x129mm (600 x 600 DPI)

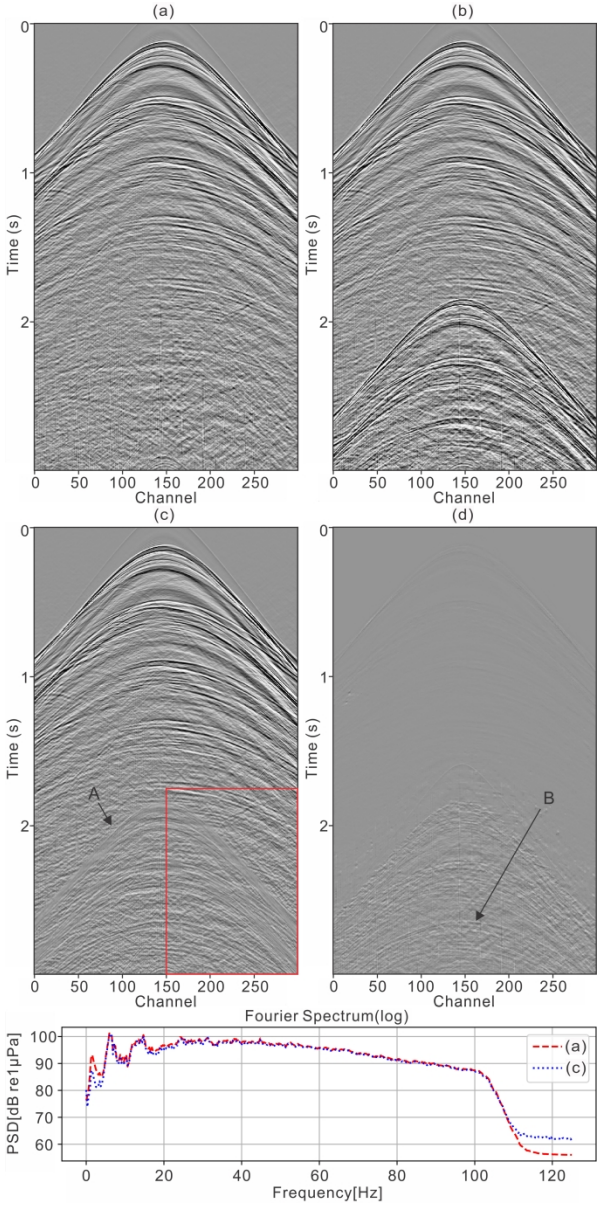


Figure 9: An example of seismic deblending by CNN after resorting to the shot domain: (a) ground truth, (b) blended data, (c) deblended data, and (d) difference between ground truth and deblended data. Arrow A indicates the blank stripes (low amplitude areas) and arrow B indicates the remaining blended signal.

109x222mm (600 x 600 DPI)

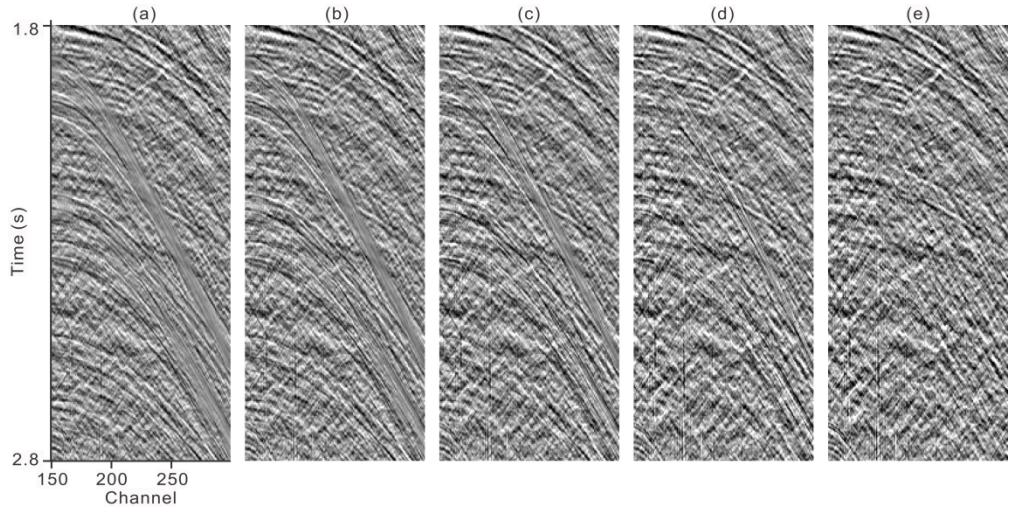


Figure 10: (a) to (d) show deblending results when 80%, 60%, 40% and 20% blending noise is introduced, respectively. (e) shows the ground truth.

168x84mm (600 x 600 DPI)

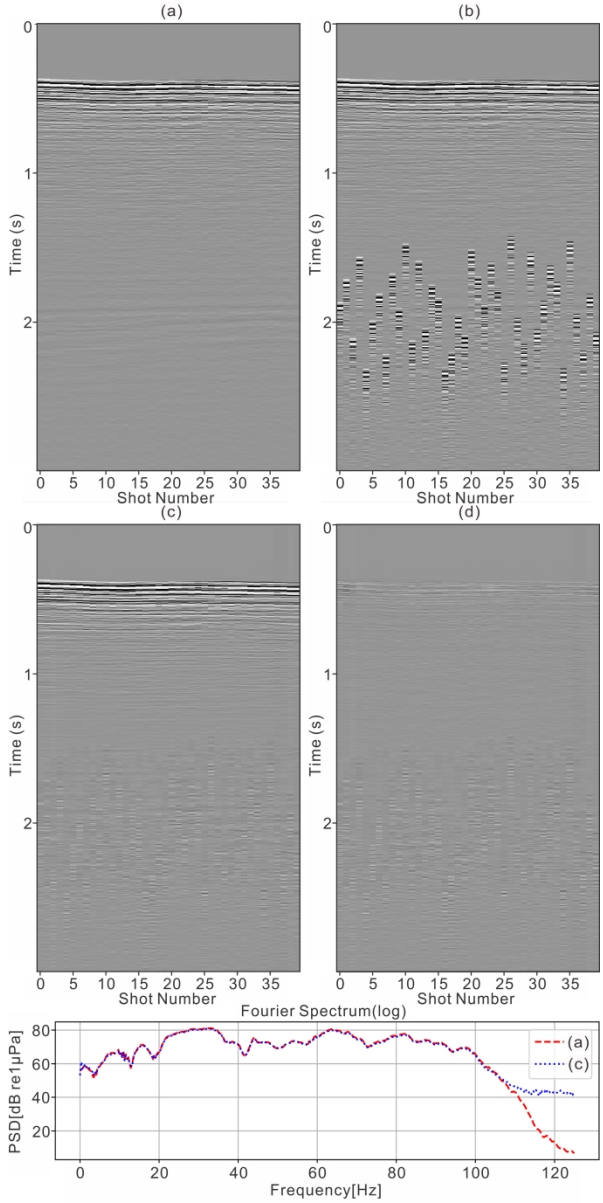


Figure 11a-d: Deblending result of applying the trained CNN on data from the North Sea in the common channel domain: (a) ground truth, (b) blended data, (c) deblended data, (d) difference between ground truth and deblended data, and (e) removed blending noise.

108x218mm (600 x 600 DPI)

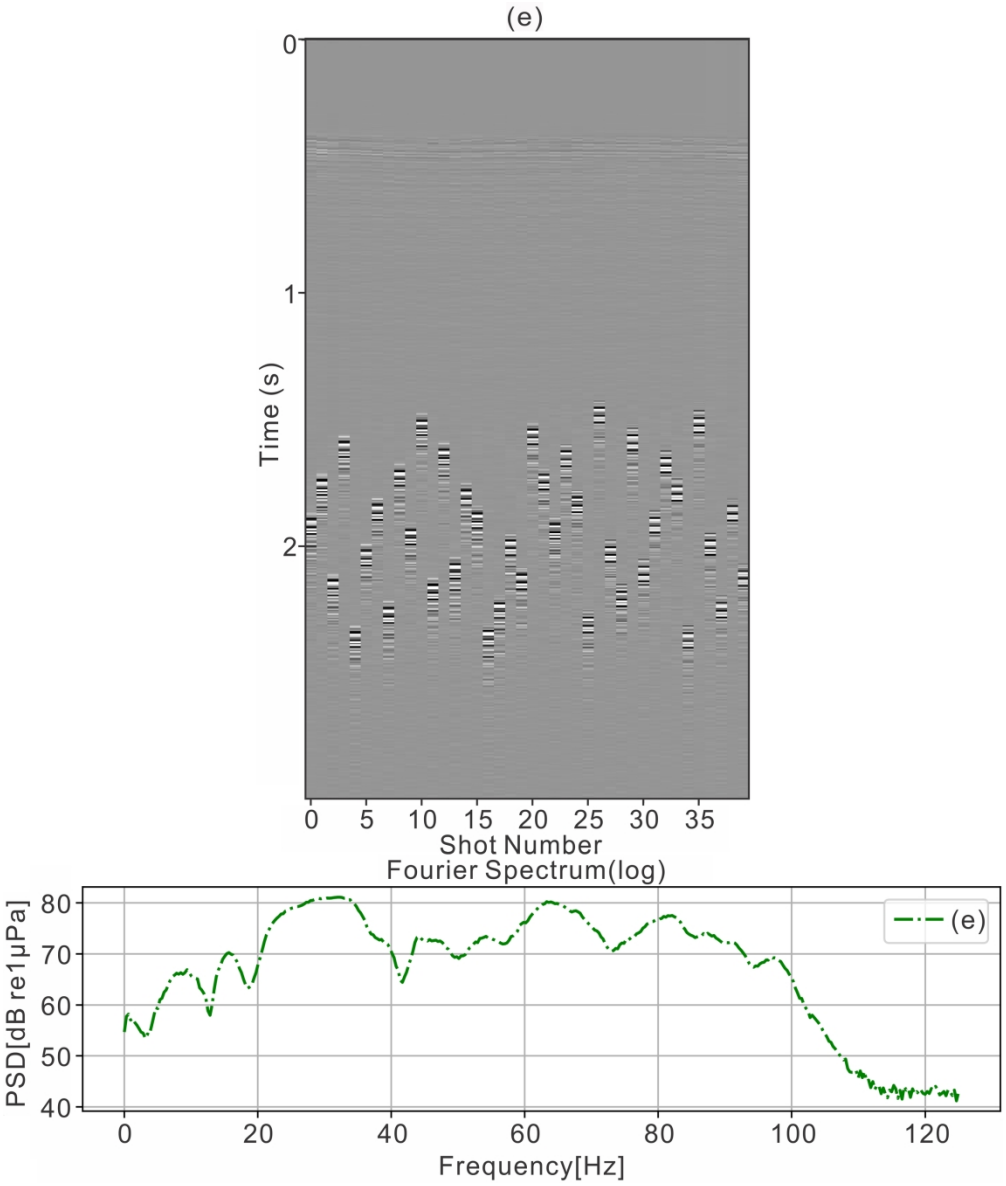


Figure 11e: Deblending result of applying the trained CNN on data from the North Sea in the common channel domain: (a) ground truth, (b) blended data, (c) deblended data, (d) difference between ground truth and deblended data, and (e) removed blending noise.

108x127mm (600 x 600 DPI)

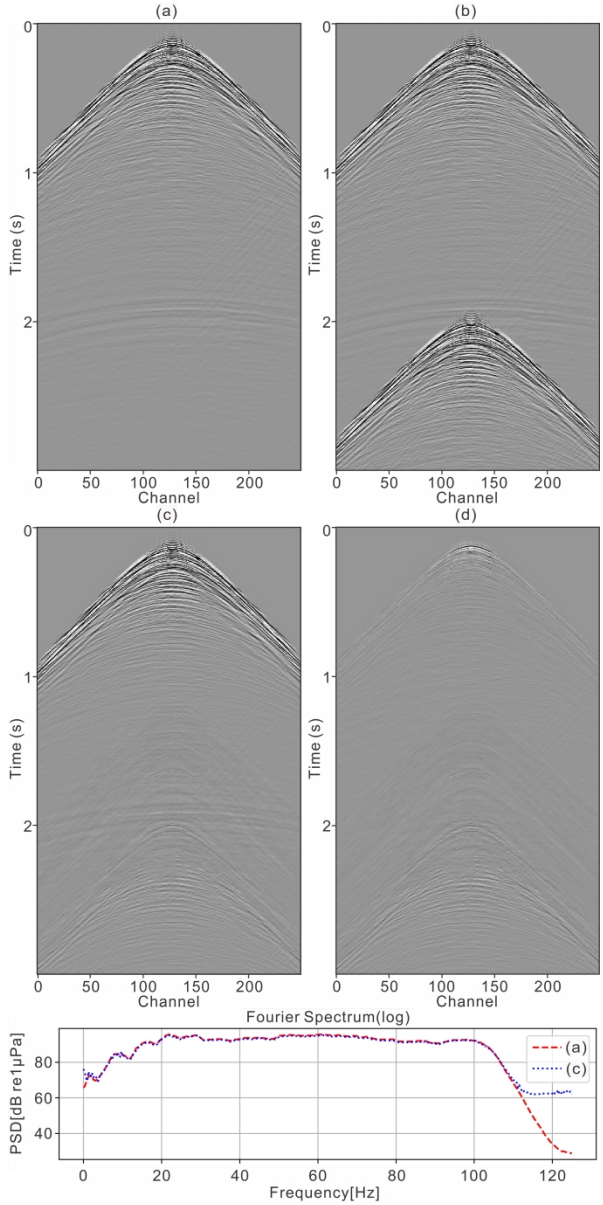


Figure 12: Deblending result of applying the trained CNN on data from the North Sea after resorting to the shot domain: (a) ground truth, (b) blended data, (c) deblended data, and (d) difference between ground truth and deblended data.

109x221mm (600 x 600 DPI)

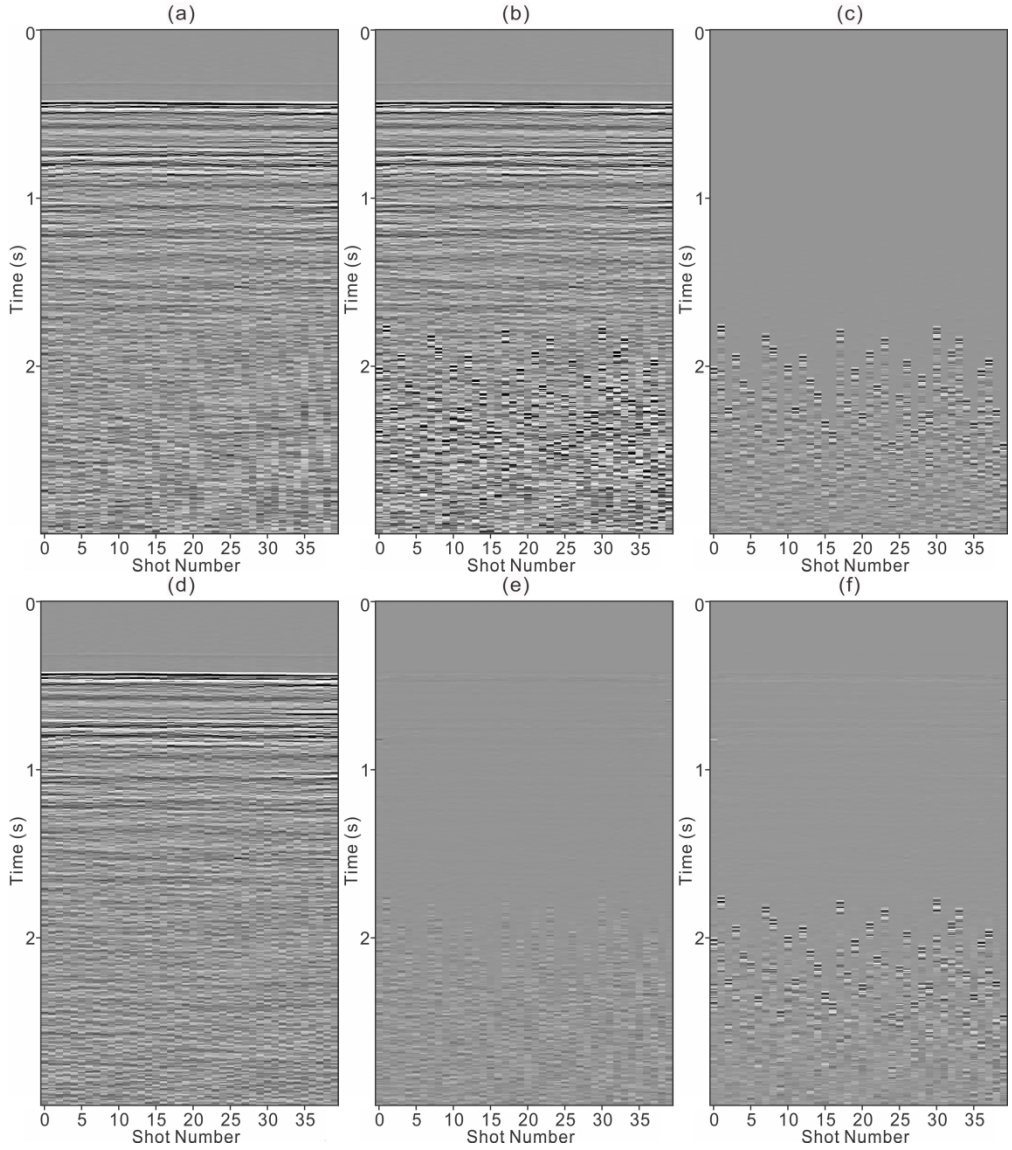


Figure 13a-f: Comparison between the CNN, F-X prediction filtering and projective filtering: (a) ground truth; (b) blended data; (c) blending noise; (d) to (f) are respectively deblended data, the difference between ground truth and deblended data, and removed blending noise employing the CNN; (g) to (i) are respectively deblended data, the difference between ground truth and deblended data, and removed blending noise employing F-X prediction filtering; (j) to (l) are respectively deblended data, the difference between ground truth and deblended data, and removed blending noise employing projective filtering.

109x125mm (600 x 600 DPI)

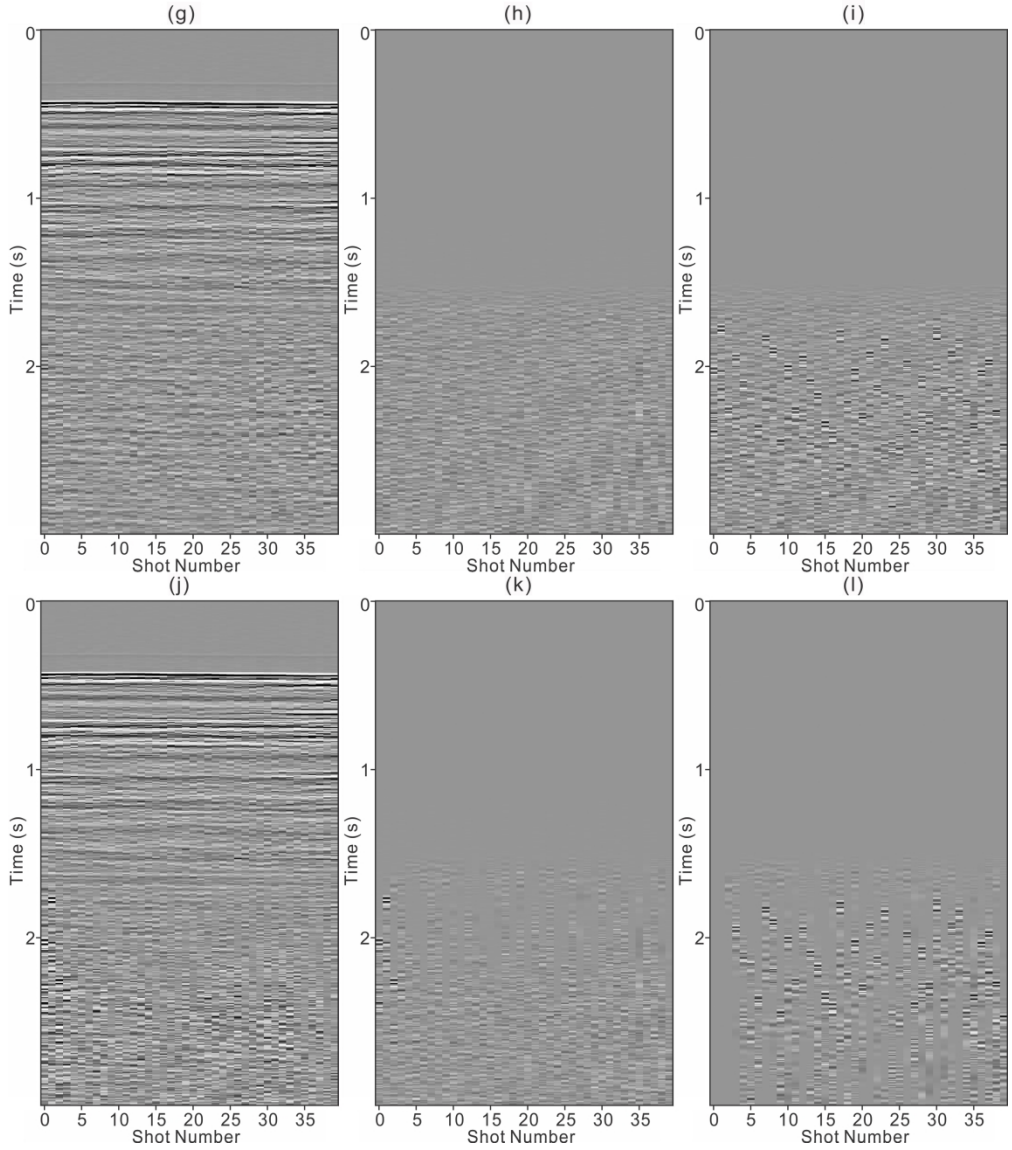


Figure 13g-l: Comparison between the CNN, F-X prediction filtering and projective filtering: (a) ground truth; (b) blended data; (c) blending noise; (d) to (f) are respectively deblended data, the difference between ground truth and deblended data, and removed blending noise employing the CNN; (g) to (i) are respectively deblended data, the difference between ground truth and deblended data, and removed blending noise employing F-X prediction filtering; (j) to (l) are respectively deblended data, the difference between ground truth and deblended data, and removed blending noise employing projective filtering.

109x125mm (600 x 600 DPI)

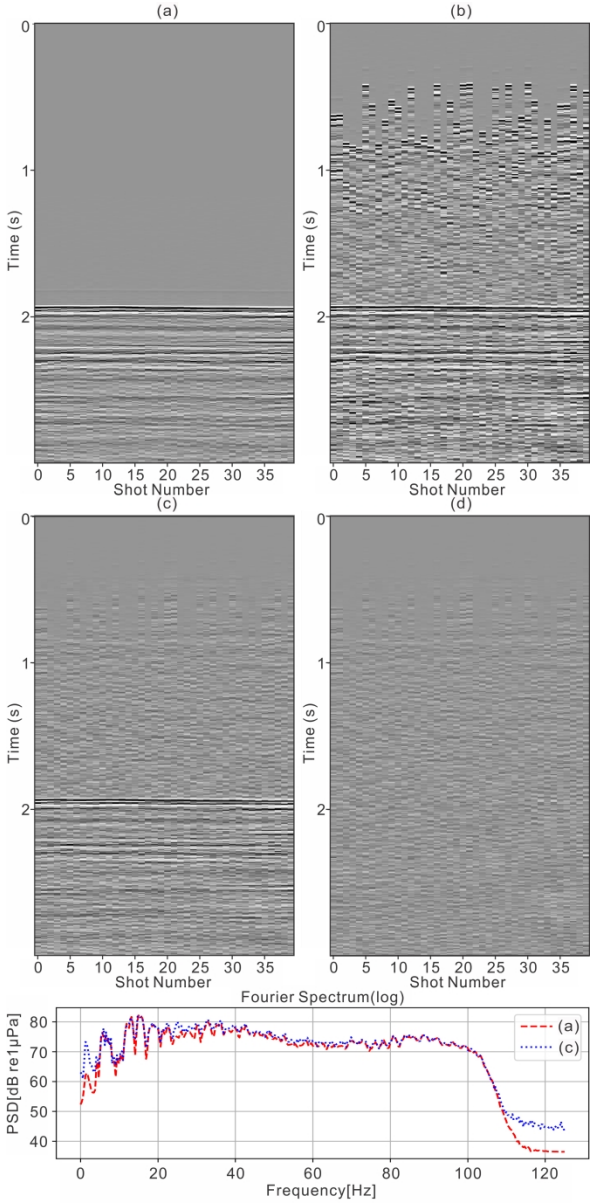


Figure A-1a-d: Deblending result of applying the trained CNN on blended data that had blending noise in the top part of the data in the common channel domain: (a) ground truth, (b) blended data, (c) deblended data, (d) difference between ground truth and deblended data, and (e) removed blending noise.

109x225mm (600 x 600 DPI)

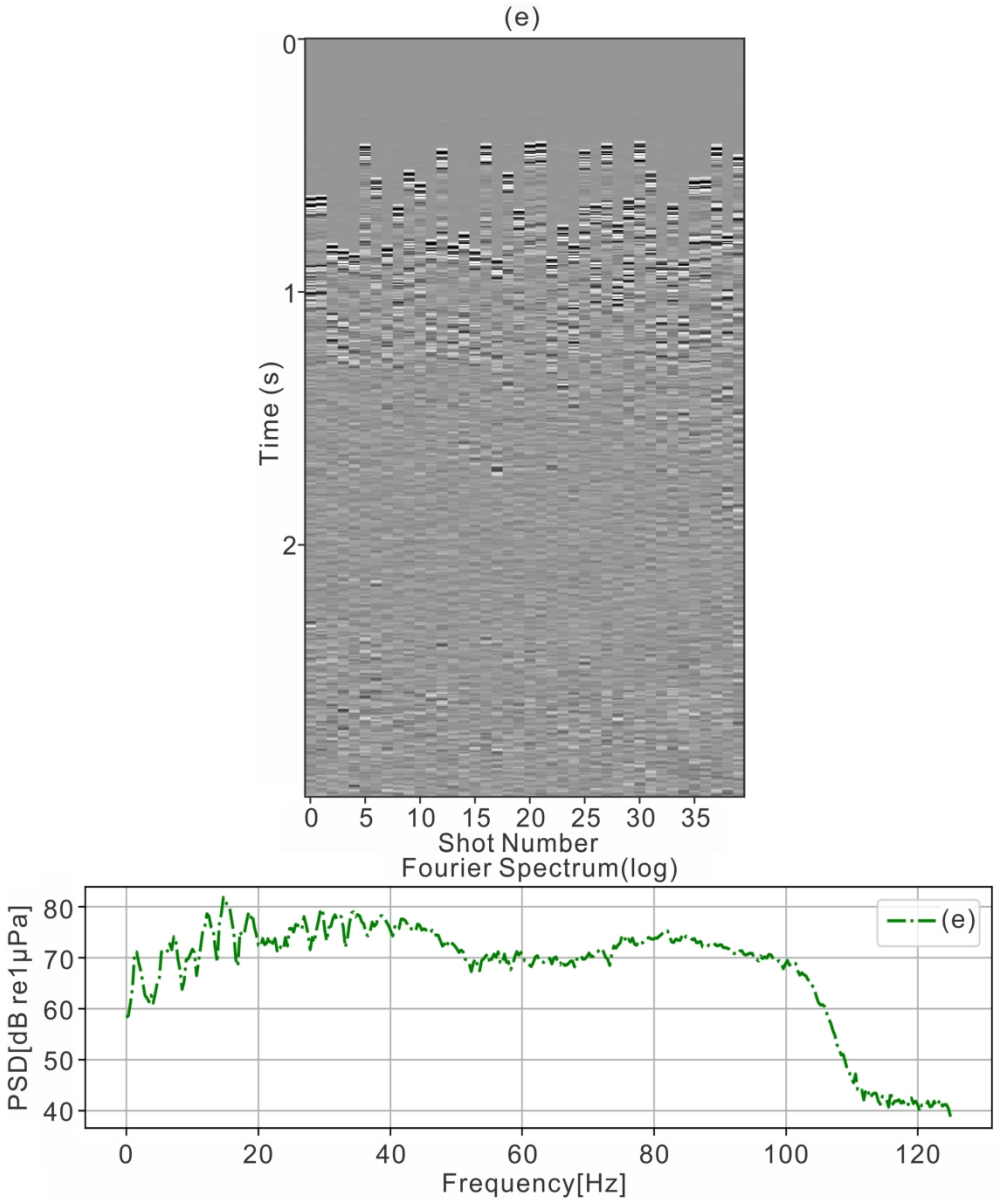


Figure A-1e: Deblending result of applying the trained CNN on blended data that had blending noise in the top part of the data in the common channel domain: (a) ground truth, (b) blended data, (c) deblended data, (d) difference between ground truth and deblended data, and (e) removed blending noise.

91x110mm (600 x 600 DPI)

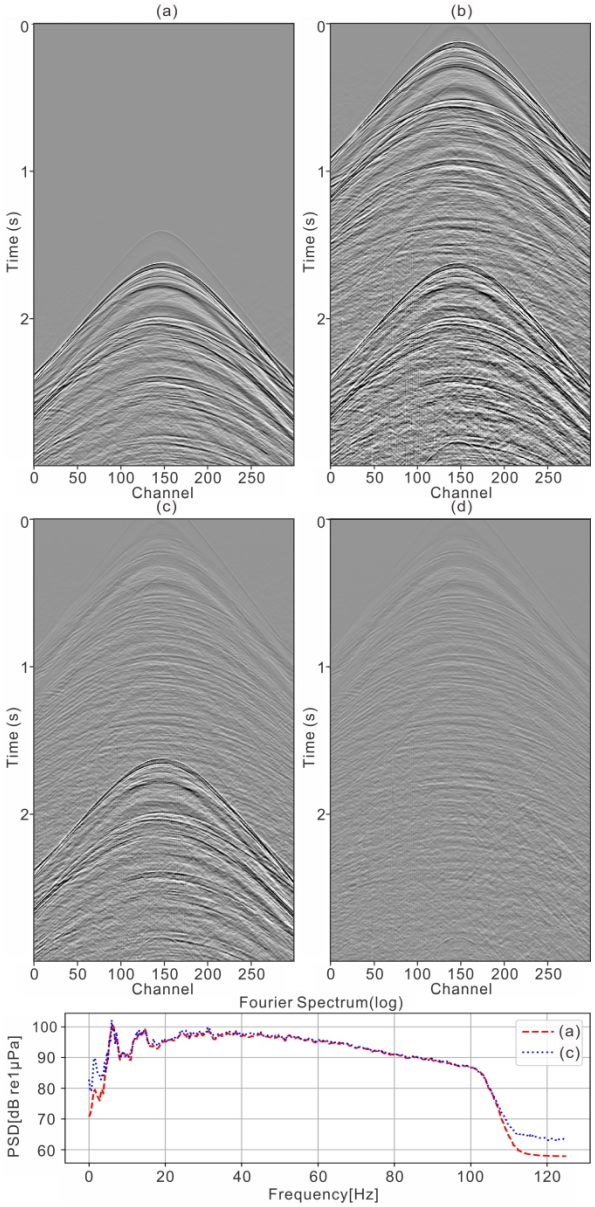


Figure A-2: Deblending result of applying the trained CNN on blended data that had blending noise in the top part of the data after resorting to the shot domain: (a) ground truth, (b) blended data, (c) deblended data, and (d) difference between ground truth and deblended data.

108x221mm (600 x 600 DPI)

Table 1: Accuracy comparison between the CNN, F-X prediction filtering and projective filtering.

| Method | CNN | F-X prediction filtering | Projective filtering |
|---|---|---|---|
| Difference with ground truth (100%) | 0.226% | 0.366% | 0.272% |
| Removed noise (blending noise=1) | 0.889 | 1.459 | 0.774 |

Table 2: Computational times of the CNN, F-X prediction filtering and projective filtering.

| Method | CNN | F-X prediction filtering | Projective filtering |
|---|---|---|---|
| Computational time (per image) | Extra 7 hours for training | 1.6s | 1.4s |
| | 0.005s | | |

## DATA AND MATERIALS AVAILABILITY

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