CS230

Least-squares seismic imaging using deep neural network inverse Hessian

Milad Bader & Júlio Oliva Frigério
Department of Geophysics
Stanford University
nmbader@stanford.edu, juliojof@stanford.edu

Abstract

Seismic imaging is used to map Earth geologic structures for energy resources exploration and characterization. Conventional methods yield blurred images with illumination artifacts. We developed a supervised convolutional neural network (CNN) approach to increase the resolution and compensate for illumination imbalance in conventional seismic images. By letting the CNN learn the appropriate mapping between input and output images, it can recover a better representation of the geologic structures.

1 Introduction

Active seismic experiments consist sending "sound" waves into the subsurface of the Earth using a human-made source (e.g. explosion, vibrator, air-gun) and recording their echo at the surface using receivers (e.g hydrophones, geophones, fiber-optic cables). A seismic echo will occur when the wave encounters an interface between any two geologic layers which have different properties. These properties are in general wave speed and medium density but the latter is often ignored. We designate by $\mathbf{m}$ the Earth model of seismic wave speed and by $\mathbf{d}$ the data collected at the surface. The data is a collection of time series, one time series per receiver, and is related to $\mathbf{m}$ non-linearily via the wave equation (PDE): $\mathbf{d} = \mathbf{d}(\mathbf{m})$. This relationship can be "linearized" and written as

$$\mathbf{d_1} = \mathbf{B}(\mathbf{m}_0)\mathbf{r}, \tag{1}$$

where $\mathbf{m}_0$ is a smooth version of $\mathbf{m}$ called background Earth model, $\mathbf{r}$ is the subsurface reflectivity characterizing the geologic interfaces, and $\mathbf{B}$ is the linearized modeling operator that depends on $\mathbf{m}_0$ and on the data acquisition parameters (e.g. source-receiver geometry, source signal bandwidth). The background Earth model is assumed known as it can be estimated using techniques such as traveltime tomography and waveform inversion. In this work, we will omit $\mathbf{m}_0$ from $\mathbf{B}$ for conciseness and we will consider fixed acquisition parameters.

The ultimate goal is to recover the best approximation of the reflectivity $\mathbf{r}$. This is the seismic imaging (inverse) problem (Claerbout, 1985). $\mathbf{r}$ has values ranging from -1 to 1 (dimensionless) and is used for subsurface structural interpretation and lithology classification to find and characterize potential oil and gas reservoirs, geothermal sources, potential sites for carbon capture and sequestration, and in many other applications.

The operator $\mathbf{B}$ in (1) is seldom explicitly built as a matrix since it becomes computationally and memory-wise prohibitive for reasonably-sized problems. Alternatively, only the matrix-vector products $\mathbf{Bv}$ and $\mathbf{B}^*\mathbf{u}$ can be computed ($\mathbf{B}^*$ is the adjoint of $\mathbf{B}$), but even these products are computationally intensive as they require solving numerically a PDE. The conventional method to

CS230: Deep Learning, Winter 2022, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

estimate the reflectivity is to use the adjoint operator only, which reads

$$\mathbf{r}_{mig1} = \mathbf{B}^* \mathbf{d}_1. \tag{2}$$

It is like replacing the deconvolution with transpose convolution in neural networks decoder. This imaging method is called "migration" (which why we use the subscript mig).

$\mathbf{r}_{mig1}$ is a degraded version of $\mathbf{r}$ because $\mathbf{B}^* \neq \mathbf{B}^{-1}$ (the inverse may not exist at all). This degradation translates into loss of resolution, reflectors dimming, and illumination artifacts; it gets worse when the background Earth model $\mathbf{m}_0$ is complex, the acquisition geometry is irregular or sparse, or the source signal has a very limited bandwidth. In this work, our goal is to use a supervised Convolutional Neural Network (CNN) that takes as input a single channel image that approximates $\mathbf{r}$ using the conventional method (e.g. $\mathbf{r}_{mig1}$), and outputs an enhanced version of that image that is "closer" to the true reflectivity $\mathbf{r}$.

2 Related work

A well-known approach that leads to a higher-resolution reflectivity image with less illumination artifacts is to build the least-squares migration (LSM) (Nemeth et al., 1999)

$$\mathbf{r}_{lsm} = (\mathbf{B}^*\mathbf{B})^{-1}\mathbf{B}^*\mathbf{d}. \tag{3}$$

LSM requires inverting the Hessian matrix $\mathbf{H}_r = \mathbf{B}^*\mathbf{B}$. This can be done using iterative solvers such as conjugate-gradient least-squares (CGLS) or GMRES but can take many expensive iterations to converge to a good solution. Torres and Sacchi (2022) used a CNN to regularize the iterative inverse problem and accelerate convergence by projecting the gradient at each iteration onto an admissible set determined by the pre-trained CNN. However, this approach is still expensive as it relies on iterative solution. Moreover, the size of the needed CNN is proportional to the number of iterations which is unknown *a priori*. Alternative methods have been proposed to replace or accelerate the LSM iterative process using matching filters that approximate the inverse Hessian matrix (Aoki and Schuster, 2009; Wang et al., 2017; Guitton, 2017). However, the effectiveness of such methods remain limited. Torres and Sacchi (2021) used a CNN to replace the matching filters in estimating the inverse of $\mathbf{H}_r$ but the training strategy relies on the true reflectivity which is never known in field applications. Thus, the method does not take advantage of field data when they are available or account for the effects of acquisition parameters and background model $\mathbf{m}_0$. Therefore, it can hardly be generalized.

3 Method

The method we propose is similar in spirit to the one suggested in Torres and Sacchi (2021), that is we aim at estimating $\mathbf{H}_r^{-1}$ using a deep CNN, $\mathcal{N}(\theta)$, but with a different training strategy. The set of network parameters are designated by θ. We call our method LSMNet and we do not use the true reflectivity $\mathbf{r}$ as CNN training labels as is the case in Torres and Sacchi (2021). Instead, the output (training labels) is a set of migrated images $\mathbf{r}_{mig1}^{(i)} = \mathbf{B}^*\mathbf{d}_1^{(i)}$, and the input is the "double-migrated" images

$$\mathbf{r}_{mig2}^{(i)} = \mathbf{B}^*\mathbf{d}_2^{(i)} = \mathbf{B}^*\mathbf{B}\mathbf{r}_{mig1}^{(i)}. \tag{4}$$

We train $\mathcal{N}(\theta)$ to learn the mapping $\mathbf{H}_r^{-1}$ between $\mathbf{r}_{mig2}^{(i)}$ and $\mathbf{r}_{mig1}^{(i)}$. Then, we use $\mathcal{N}(\theta)$ in inference mode to obtain an enhanced reflectivity image

$$\mathbf{r}_{cnn} = \mathcal{N}(\mathbf{r}_{mig1}; \theta). \tag{5}$$

We would like $\mathbf{r}_{cnn}$ to be "better" than the traditional migration $\mathbf{r}_{mig1}$ by being "closer" to the true reflectivity $\mathbf{r}$. Our approach has two main advantages compared to using $\mathbf{r}$ as training labels. Firstly, having migrated images on both ends of the CNN makes the latter more agnostic to the background Earth model $\mathbf{m}_0$ and acquisition parameters which are implicitly embedded non-linearily in the operator $\mathbf{B}$ used to generate the migrated images. Secondly, when field data is available (as opposed to synthetic data), it can also be used to augment the training dataset. This cannot be done when using $\mathbf{r}$ in the CNN training since it is never known in reality. These advantages make our approach more generalizable and robust. However, it remains "under-ambitious" in trying to recover the true reflectivity $\mathbf{r}$ since it is unable to fill the null-space of the Hessian. As an alternative, we also tried the

approach where we train the CNN using both $\mathbf{r}_{mig1}$ and $\mathbf{r}_{mig2}$ as two input channels and $\mathbf{r}$ as labels directly. We write the prediction formally as

$$\mathbf{r}_{cnn} = \mathcal{N}(\mathbf{r}_{mig1}, \mathbf{r}_{mig2}; \theta). \tag{6}$$

With this second approach, the network is not strictly approximating the inverse Hessian anymore and we cannot train it with field data when available. Nevertheless, we maintain the first advantage mentioned above since the non-linearily in the operator $\mathbf{B}$ is still embedded in the relation between the two channels $\mathbf{r}_{mig1}$ and $\mathbf{r}_{mig2}$.

There is no CNN model available to serve as a baseline for our task. However, we use the architecture from Torres and Sacchi (2021) to build and train our benchmark CNN model. This architecture is illustrated in Figure 1. It is a lighter version of U-net with 5 convolutional blocks with ReLU activation and one last convolution layer to capture the image features. Max-pooling halves the size of the image in the encoding part and the transpose convolution doubles it in the decoding part. Multiple skip connections are used to prevent gradients vanishing and transfer image features between encoding and decoding layers. The top level skip connection adds back the input image to the network output. For the first approach, we train the CNN by minimizing over θ the pixel-to-pixel misfit between $\mathbf{r}_{mig1}$ and the network output (mean squared error MSE for the benchmark model). For the second approach, we use a one-level deeper network and minimize the misfit between $\mathbf{r}$ and the network output.

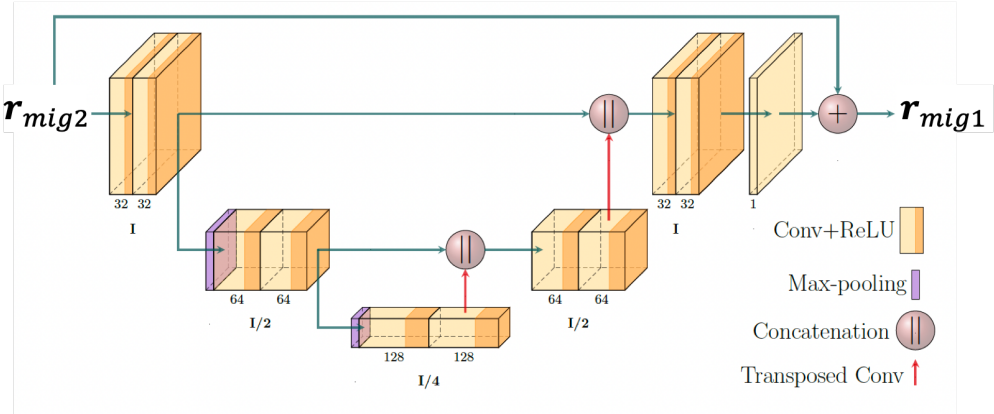


Figure 1: Benchmark CNN architecture for LSMNet (adapted from Torres and Sacchi (2021)). The number of channels is indicated below each block. The size of the input image going through each block is given with respect to the input size $\mathbf{I}$.

4 Dataset

We generated training/validation/testing images following these steps:

- Use a synthetic geologic model generator (Clapp, 2018) to build 1024 Earth models $\mathbf{m}$ of size 512 x 256 each
- Decompose each model $\mathbf{m}$ into background $\mathbf{m}_0$ and reflectivity $\mathbf{r}$
- Generate a first set of seismic data $\mathbf{d}_1$ (recordings at the surface of the Earth) solving (1)
- Compute the first migrated image $\mathbf{r}_{mig1}$ solving (2)
- Generate a second set of seismic data $\mathbf{d}_2 = \mathbf{B}\mathbf{r}_{mig1}$
- Compute the second migrated image $\mathbf{r}_{mig2}$ solving (4)

To build the Earth models, we constructed a Jupyter Notebook inside a ready-to-use Docker image which includes all the necessary Python modules (Farris, 2022). For the computations involving $\mathbf{B}$ and $\mathbf{B}^*$, we used our own open-source C++ code (Bader, 2022). We augment our dataset by flipping horizontally and splitting into 4 quadrants. We shuffle the images to randomize their order and normalize their amplitudes individually so that values fall between -1 and 1. This results in

ready-to-use 8192 pairs of images of size 256 x 128 each. We keep 512 pairs for validation (6.25%) and another 512 for testing.

Figure 2 illustrates the dataset creation workflow with an actual example. The seismic data is acquired using 11 sources and 489 hydrophones at the surface (at Depth = 0 km). Only a single source (red star) is overlaid on top of $\mathbf{m}$ for simplicity, along with the corresponding hydrophones (yellow horizontal line). The seismic recordings ("echos") $\mathbf{d}_1$ and $\mathbf{d}_2$ from that same source are also shown. Red (green) arrows indicate the application of the operator $\mathbf{B}$ ($\mathbf{B}^*$). The black dashed boxes indicate the datasets used in the CNN training for our first approach. Notice how the traditional image $\mathbf{r}_{mig1}$ is a blurred version of the true reflectivity $\mathbf{r}$ with loss of continuity due to the imbalanced illumination. Notice also the acquisition footprints visible in particular at the first shallow reflector (horizontal "beating" at the top of the image). The second migrated image $\mathbf{r}_{mig2}$ gets even worse because of the compounded effects of successively applying the operator $\mathbf{B}^*$.

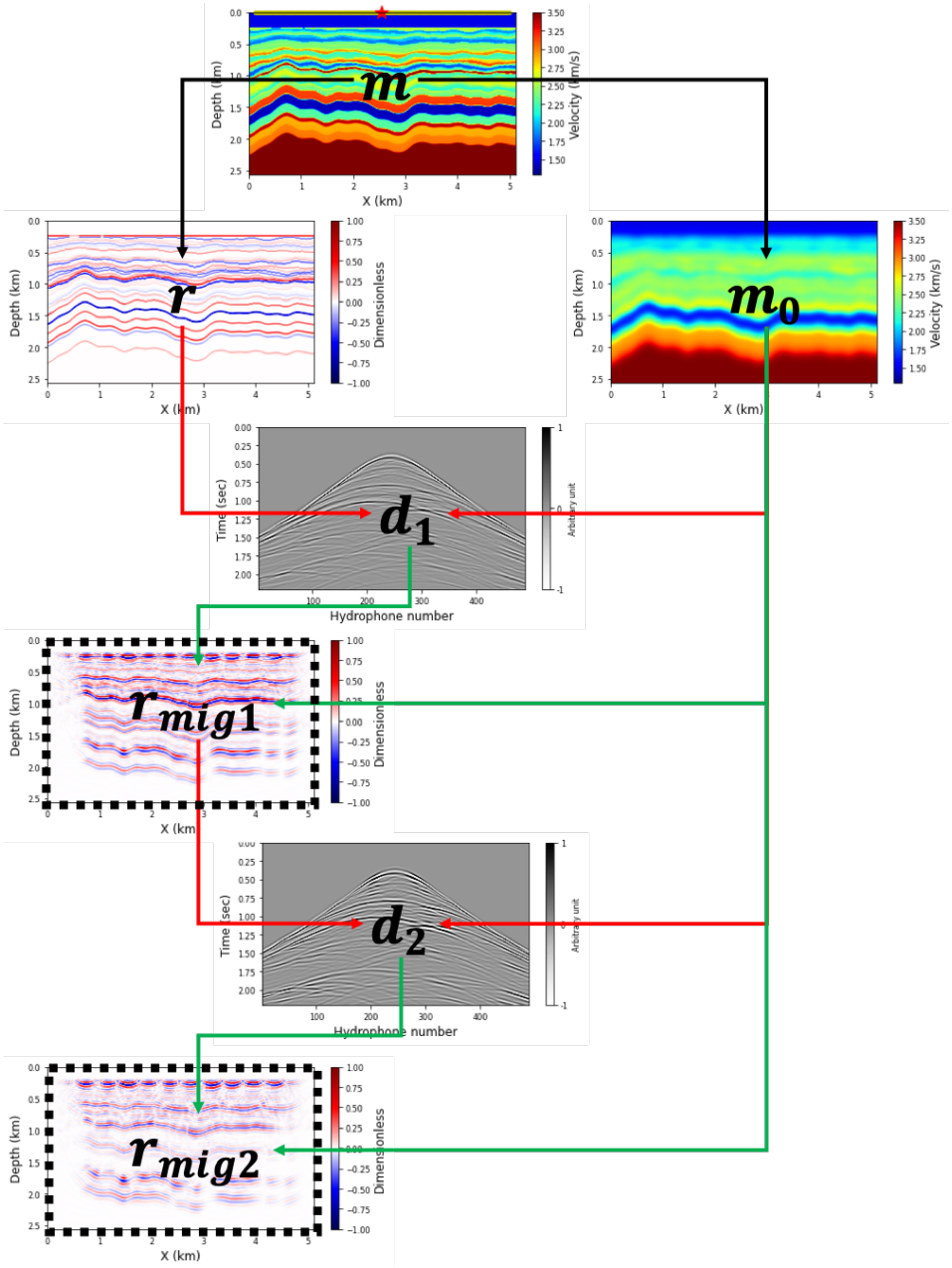


Figure 2: Diagram illustrating the workflow for generating the dataset for LSMNet.

Parameter	Benchmark	Upgrade	Notes
Activation	ReLU	leaky ReLU	Allow negative values
Normalization	None	Batchnorm	Improve convergence
Loss	MSE	l_1	More attention to weaker reflections
Mini-batch size	2	16	Regularization effect
Learning rate	0.001	0.005	Faster training
Other	-	$tanh$ activation before the output	Constrain output to [-1,+1]

Table 1: Difference between benchmark and upgraded CNN.

5 Experiments and Results

In this section, we cover the first approach described in Section 3. Refer to the video-presentation for the results with the second approach. As mentioned previously, we trained a benchmark CNN model following Torres and Sacchi (2021) whenever possible. Then, we upgraded the model by changing certain (hyper)parameters in order to improve the final results. Since training is computationally intensive, we did not iterate on every parameter separately but we changed a bundle of parameters at once. Since we deal with an image-to-image problem, our measure of performance/effectiveness was the cost function itself, the cosine similarity metric between images (e.g. CNN output and labels) $cs = \frac{image_1^T image_2}{\|image_1\|\|image_2\|}$, and the qualitative assessment of the sought seismic image $\mathbf{r}_{cnn}$ compared to the conventional one $\mathbf{r}_{mig1}$. We select the best CNN model and detail its differences compared to the benchmark in Table 1.

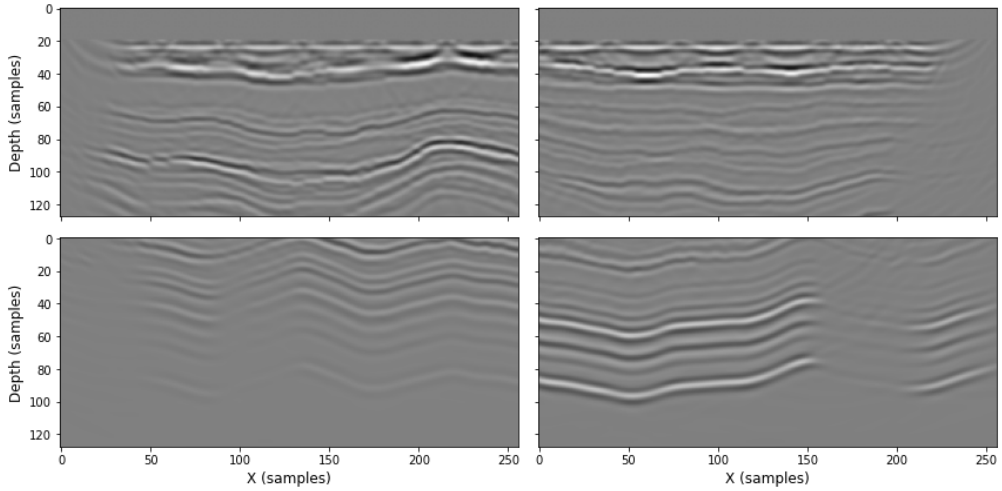
We train both models from scratch with 21 epochs. We show in Appendix (Figures 4 and 5) the cost functions and cosine similarity metric for both models during training and validation. Once trained, we use each model to predict (infer) the reflectivity according to equation 6 for all 512 testing examples. The corresponding cosine similarity metric is also shown in Appendix (Figure 6). Generally, both CNN models perform well in terms of convergence and metric. However, looking only at the global measures above may lead to conclude that the benchmark model performs better. Figure 3 shows a comparison between conventional image $\mathbf{r}_{mig1}$ and predicted reflectivity using both models for 4 examples from the testing set. Figure 7 in Appendix shows the corresponding true reflectivity $\mathbf{r}$. We observe an improvement in the CNN predicted images (illumination compensation, reflectors continuity, and vertical resolution) compared to $\mathbf{r}_{mig1}$. The improvement is more pronounced when we use the upgraded CNN. Figure 8 in Appendix shows the average amplitude spectra of the different seismic images. The spectra clearly show that the upgraded CNN was able to improve the vertical resolution of the images (wider wavenumber spectrum) more than the benchmark CNN. We can deduce from these observations that using pixel-to-pixel comparison may not be the optimal measure of performance or the best type of loss to use in training the CNN with seismic images.

6 Conclusion and Future Work

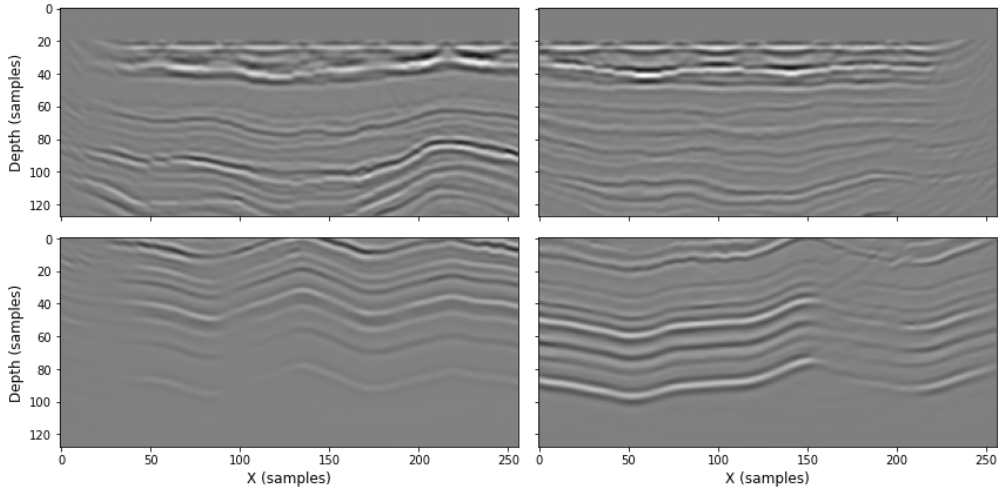
We have developed a novel method based using a CNN with a U-net architecture to estimate the inverse Hessian relating pairs of seismic images. We used our network in inference mode to predict Earth subsurface reflectivity. Our method leads to an improved image compared to conventional methods. The results quality depend on CNN parameters and training strategy, and global performance metrics may be misleading if not investigated carefully. For future work, we would like to use more sophisticated methods to measure similarity between seismic images such as the structural similarity index measure (SSIM) (Wang et al., 2004) or other relevant measures. We would like also to train our CNN using more sophisticated (and common) Earth models including faults and salt bodies, and apply our method to field seismic images.

7 Contributions

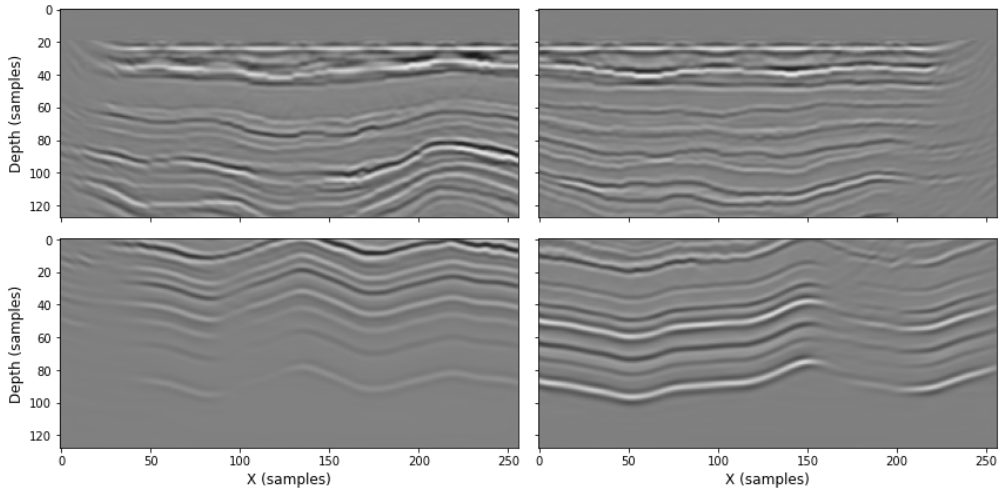
Milad generated the synthetic Earth models along with the data depicted in Figure 2. Julio built the benchmark CNN and tested it on a batch of examples. Both Milad and Julio built subsequent models starting from the benchmark CNN.



(a) Conventional images $\mathbf{r}_{mig1}$



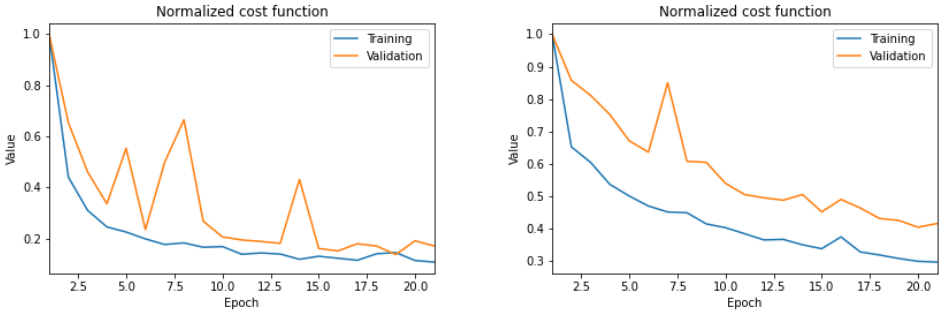
(b) Predicted images by benchmark CNN



(c) Predicted images by upgraded CNN.

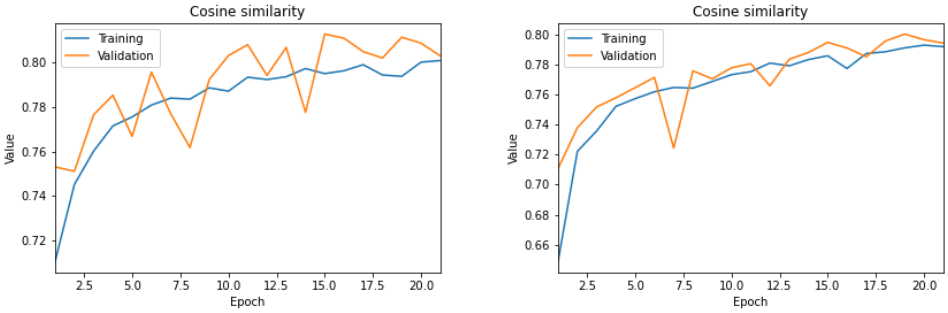
Figure 3: Four examples of seismic images from the testing set.

8 Appendix



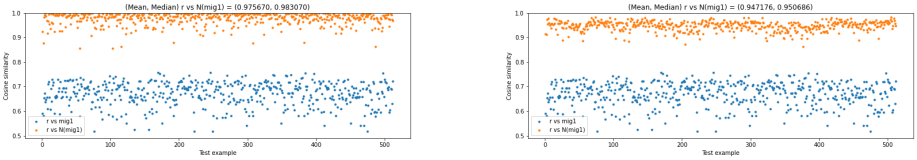
(a) Benchmark CNN (b) Upgraded CNN

Figure 4: Cost function during the training of the CNN models. (a) MSE. (b) l_1-norm.



(a) Benchmark CNN (b) Upgraded CNN

Figure 5: Average cosine similarity between labels and CNN output during the training of the CNN models.



(a) Benchmark CNN (b) Upgraded CNN

Figure 6: Cosine similarity between true reflectivity $\mathbf{r}$ and migrated image before and after using the CNN inference on all 512 testing examples.

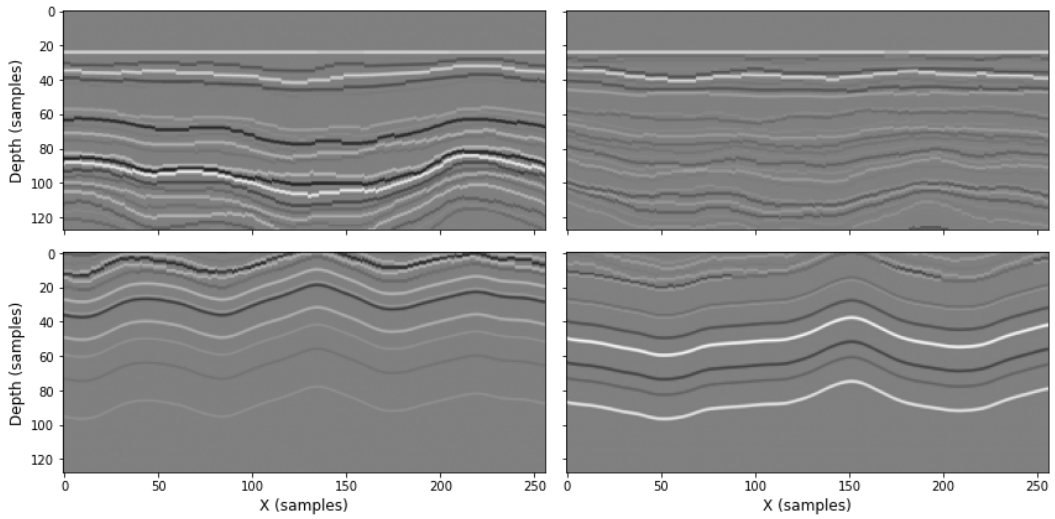


Figure 7: True reflectivity corresponding to the examples shown in Figure 3.

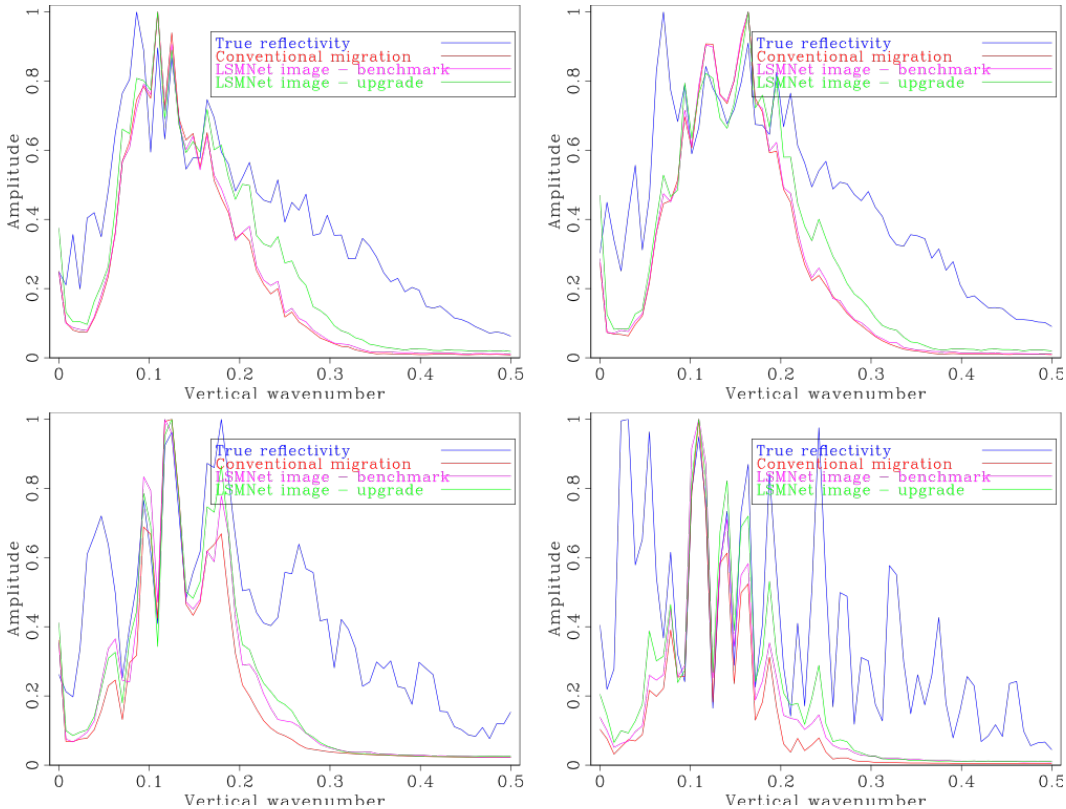


Figure 8: Average amplitude spectra corresponding to the seismic images shows in Figures 3 and 7.

References

Aoki, N. and Schuster, G. T. (2009). Fast least-squares migration with a deblurring filter. *Geophysics*, 74(6):WCA83–WCA93.

Bader, M. (2022). fwi2d. `https://github.com/nmbader/fwi2d`.

Claerbout, J. F. (1985). *Imaging the earth's interior*, volume 1. Blackwell scientific publications Oxford.

Clapp, R. G. (2018). Synthetic model building for training neural networks in a Jupyter notebook. *SEP-Report*, 172.

Farris, S. (2022). pysynth. `https://hub.docker.com/r/sfarris1994/pysynth`.

Guitton, A. (2017). Fast 3D least-squares RTM by preconditioning with nonstationary matching filters. In *2017 SEG International Exposition and Annual Meeting*. OnePetro.

Nemeth, T., Wu, C., and Schuster, G. T. (1999). Least-squares migration of incomplete reflection data. *Geophysics*, 64(1):208–221.

Torres, K. and Sacchi, M. (2021). Deep learning based least-squares reverse-time migration. In *SEG/AAPG/SEPM First International Meeting for Applied Geoscience & Energy*. OnePetro.

Torres, K. and Sacchi, M. (2022). Least-squares reverse time migration via deep learning-based updating operators. *Geophysics*, 87(6):S315–S333.

Wang, M., Huang, S., and Wang, P. (2017). Improved iterative least-squares migration using curvelet-domain Hessian filters. In *2017 SEG International Exposition and Annual Meeting*. OnePetro.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.