



Enriching synthetic data with real noise using Neural Style Transfer

Naomi Takemoto^{1,2*}, Lucas de M. Araújo^{1,2}, Tiago A. Coimbra¹, Martin Tygel¹, Sandra Avila^{1,2}, and Edson Borin^{1,2}

¹Center for Petroleum Studies (CEPETRO), University of Campinas (Unicamp), Brazil

²Institute of Computing (IC), University of Campinas (Unicamp), Brazil

Copyright 2019, SBGf - Sociedade Brasileira de Geofísica.

This paper was prepared for presentation at the 16th International Congress of the Brazilian Geophysical Society, held in Rio de Janeiro, Brazil, August 19-22, 2019.

Contents of this paper were reviewed by the Technical Committee of the 16th International Congress of The Brazilian Geophysical Society and do not necessarily represent any position of the SBGf, its officers or members. Electronic reproduction or storage of any part of this paper for commercial purposes without the written consent of The Brazilian Geophysical Society is prohibited.

Abstract

Deep Learning experiments require large amounts of labeled data, but few annotated seismic datasets are available and annotation is a time-consuming, expensive activity. Synthetic modeled datasets may be a viable alternative. However, they lack the variability and intricacies of a real data signal. Moreover, methods that add colored noises are not enough to represent such variability. Thus, our goal is to produce a noise type that is characteristic of real data to create a viable synthetic dataset to train Deep Learning models. In this context, we apply the Neural Style Transfer technique, which combines the structural content of an image with the textural style of another, to produce a synthetic data with noise characteristics extracted from a real dataset. The results show that the stylized synthetic seismic data preserves the modeled content while incorporating characteristics of some real data chosen as style, creating synthetic data with a more realistic noise profile.

Introduction

In recent years, we have observed a steep increase in Deep Learning (DL) research applied to geophysical problems, such as the usage of Deep Neural Networks (Araújo et al., 2018), Convolutional Neural Networks (Wu et al., 2018), and Generative Adversarial Networks (Lu et al., 2018). In the context of the seismic data, works are emerging in data processing and interpretation, related to topics such as noise removal, data interpolation, seismic fault detection, fascias classification, and several others.

Although DL techniques have been successful in increasingly complex tasks, the necessary amount of data to train such systems is very high, typically on the order of tens of thousands for proof-of-concept research and millions of training examples for more robust, industrial applications. As discussed by Saxena (2018), one of the challenges for DL research in this field is that seismic data are abundant but very little of it is appropriately labeled. Mainly due to its costly nature, requiring manual, specialized labor to annotate each data set.

As the scarcity of labeled data sets is an obstacle in

applying DL tools to seismic data, the use of synthetic data presents itself as an alternative to this limitation. Many seismic phenomena have well understood and accepted mathematical models, and usually these models can be used to generate synthetic data. Since the model is known, synthetic data are labeled by construction. However, the apparent drawback of synthetic data is that it lacks the feature richness of real seismic data. In other words, if on the one hand, we may be able to perform proof-of-concept experiments, on the other quickly we may overestimate how well a DL system is adequate to solve a problem in a real-world situation.

Two common strategies in adding uncertainty to synthetic data are the addition of colored noise (typically some Gaussian distribution within a particular frequency bandwidth) and the superposition of a noise-only section of real data (see Brenders and Dellinger (2016), for example). The limitation of the first strategy is that it does not capture all the nuances of real data noise. The limitation of the second strategy is that typically only deep marine data have large enough noise-only sections, so other types of data cannot be aptly represented.

This work investigates a different approach to the problem: we propose to use the Neural Style Transfer technique to enrich synthetic data with real seismic data characteristics. In other words, given a synthetic data S and a real data R , we produce a new data that has the structure profile of S (what was modeled) with the feature profile of R .

Neural Style Transfer

Neural Style Transfer (NST) is a technique that combines the content of an image with the style of another to produce a novel result, such as the one presented in Figure 1. This section presents the NST technique, in the color image context of the original paper (Gatys et al., 2016).

Typically, we represent a digital color image by three color channels, red, green and blue (RGB). Convolutional Neural Networks learn to extract features from this raw representation in a hierarchy of abstractions, with the first layers learning simple features such as edge or texture detection and the latter layers learning increasingly abstract features, up to classification of different objects (Lee et al., 2009). This hierarchical abstraction allows for object recognition independent of characteristics such as color, size, orientation, and framing.

Gatys et al. (2016) showed that we can correlate these layers with information about content and style in an image. This is the insight that led to the Neural Style Transfer technique, which creates an image that both correlates its content with one image and its style with another.



Figure 1: Example of the Neural Style Transfer technique. Figure reproduced from Gatys et al. (2016).

To implement the technique, the authors used a pre-trained VGG network (Simonyan and Zisserman, 2014) as a feature extractor.

Definitions

In this section, we present the Neural Style Transfer's loss function as presented in Jing et al. (2017). NST is essentially an optimization problem which the goal is to generate an image I that combines the content of a reference image I_C with the style of another image I_S . Lets define:

- Feature map: $F^l(I) \in \mathbb{R}^{C \times H \times W}$ is the output volume of VGG's layer l , with dimensions C (number of channels), H (height) and W (width).
- Reshaped feature map: $F_R^l(I) \in \mathbb{R}^{C \times HW}$ is the output of the linearization of each $H \times W$ channel to a HW size vector.

Given the content and style images (I_C and I_S) and respective feature and reshaped feature maps, we can define I^* as the image that best combines content and style, minimizing

$$\begin{aligned} I^* &= \arg \min_I \mathcal{L}_{total}(I_C, I_S, I) \\ &= \arg \min_I \alpha \mathcal{L}_C(I_C, I) + \beta \mathcal{L}_S(I_S, I). \end{aligned} \quad (1)$$

In Equation 1, α and β are the hyperparameters that balance the content preservation vs stylizing trade-off.

The content cost \mathcal{L}_C is defined as the sum of the Euclidean distances between the feature maps of each layer of the generated and content images:

$$\mathcal{L}_C = \sum_{l \in \{l_c\}} \|F^l(I_C) - F^l(I)\|^2, \quad (2)$$

where l is a layer in l_c , which is the set of layers chosen to correlate with the content features.

The style cost is based on the Gram matrix of the reshaped feature maps, where each entry $G_{i,j}$ is the scalar product between columns i, j of a given matrix. In this context, the Gram matrix correlates color and texture without regard

to the global arrangement. The Gram matrix for each reshaped feature map is computed as

$$G(F_R^l(I)) = [F_R^l(I)][F_R^l(I)]^T. \quad (3)$$

The style cost is then defined as

$$\mathcal{L}_S = \sum_{l \in \{l_s\}} \|G(F_R^l(I_S)) - G(F_R^l(I))\|^2. \quad (4)$$

Again, the optimization's goal is to minimize the total cost, defined as a linear combination of \mathcal{L}_S and \mathcal{L}_C :

$$\mathcal{L}_{total} = \alpha \mathcal{L}_C + \beta \mathcal{L}_S. \quad (5)$$

Stylization process

The Neural Style Transfer technique might be summarized as follows:

1. Choose the set of layers that will be used to compute each feature map. It is worth to notice that the style's set of layers might be different from the content's.
2. Forward propagation: feed the convolutional network with style and content images in order to obtain all the desired feature maps.
3. Initialize the generated image with the content image or with random noise.
4. Repeat the forward propagation, now using the generated image, compute its feature maps and calculate \mathcal{L}_C , \mathcal{L}_S and total loss \mathcal{L}_{total} .
5. Update the generated image (this is the optimization step).
6. Repeat steps 4 and 5 until some criteria for ending the optimization process is met (typically a fixed set of iterations).

The criteria used to determine which layers to pick for style's and content's sets influences the outcome. Particularly, the choice of deeper layers for the content results in higher levels of distortions in the generated image in contrast with choosing shallower layers.

In Figure 1 we illustrate an example of Neural Style Transfer from the original paper.

Methodology

Our goal is to adapt the NST technique for seismic data instead of images. Verma and Smith (2018) presented a method for synthesizing sounds with Neural Style Transfer. They illustrated the method applying a tuning fork timbre (style) to a melodic harp line (content) and a violin timbre (style) to a vocal melodic line (content), effectively showing that the Neural Style Transfer technique can both remove and add spectral components, respectively. Hence, they demonstrated that the technique can also be applied to other types of signals than image. In this section, we present how the input and output for the NST were transformed to work directly with seismic data.

Input data specifications

The convolutional neural network VGG requires as input an image with minimum size of 224×224 pixels and three color channels (BGR or RGB), but, aside from memory constraints, there is no upper bound since only the convolutional layers are used. Therefore, content, style, and stylized images must fit these specifications, but they do not need to have the same sizes. In this type of image representation, each input pixel is a combination between the intensity of each channel, blue (B), green (G) and red (R) and these values vary in the discrete range of $[0, 255]$.

Before the stylization process begins, it is necessary to pre-process the input image so that each channel has an approximate mean value equals to zero. The VGG network was trained on the ImageNet dataset (Deng et al., 2009) and centering was done by subtracting the ImageNet mean of each channel (Gatys, 2017). The final range of each channel is $[-123, 131]$ for red, $[-116, 138]$ for green and $[-103, 158]$ for blue. Our seismic data will have to be scaled to occupy approximately these ranges.

Pre-processing seismic data

The 2D seismic data that we used in this study are different from the expected input described above. To fit VGG input requirements, we made the following transformations:

1. First, we construct a 3D volume by repeating the seismic data in each color channel. In this process, each sample s with amplitude a is converted into a pixel with RGB values equal to a ;
2. Then, we normalize each channel in order to fit each corresponding channel range;
3. Finally, we use the previous' step outcome as input to the convolutional neural network.

Stylizing synthetic data

Given some synthetic data as content and some real seismic data as style, the stylization process is as follows:

1. Apply some noise to the synthetic data. This is important because the NST technique does not seem to apply the style in regions where the content is blank (more on that in the Results section);

2. Pre-process content and style data with the pre-processing steps above;
3. Initialize the stylized (target) data with the content data or with random noise (both approaches may yield good results, as discussed in the original paper);
4. Apply the Neural Style Transfer technique, solving the optimization problem;
5. Post-process the stylized data by rescaling each channel to some desired range $[-1, 1]$, for example) and collapse the three channels into one by computing their average.

An additional step made after stylizing the data was to sum it with the original content data. In this way, we can highlight the original structures and control the amount of noise applied in the synthetic data. Figure 2 illustrates the pipeline of the whole process.

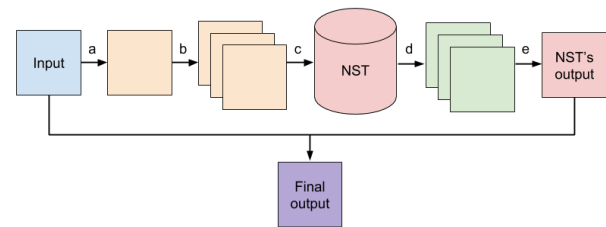


Figure 2: Diagram of the stylization process applied to seismic data. In step *a* we add Gaussian noise to the synthetic input. In *b*, we construct a 3D volume, repeating and rescaling *a*. In *c*, we submit the previous output to the Neural Style Transfer's procedures, which also receives some real data as style (omitted). In *d* we have the Neural Style Transfer's output. In *e* we collapse the 3D volume to bring the data back to the initial seismic domain data. Finally, in *f*, we add the stylized data to the initial input, choosing a desired signal-to-noise ratio.

Example

We applied the stylization process using the real dataset extracted from the Tacutu basin as style, presented in Figure 3 alongside its $f-k$ spectra, to a section of a synthetic data named Sigsbee (Figure 4a), which was stacked with the Common-Reflection-Surface (CRS) method (see, e.g., Facciopieri et al., 2016). Observing the resultant images, we see that the style tended to be transferred to similar areas (in luminosity terms), with the background not receiving as much noise as the areas with events, as in the upper left area of Figure 5a. To test this hypothesis, we added Gaussian noise to the synthetic example before applying NST. By doing this, we expected to distribute the luminosity uniformly along the seismic zero-offset (ZO) stack, thus allowing the stylization to happen beyond regions with events, as can be noted by comparing both images.

The hyperparameters used to balance the stylization vs content preservation trade-off were set in $\alpha = 1$, $\beta = 1000$ for 500 iterations. These values are the same as the ones suggested by Gatys et al. (2016), chosen after visual and empirical experimentation. The outcome is shown in

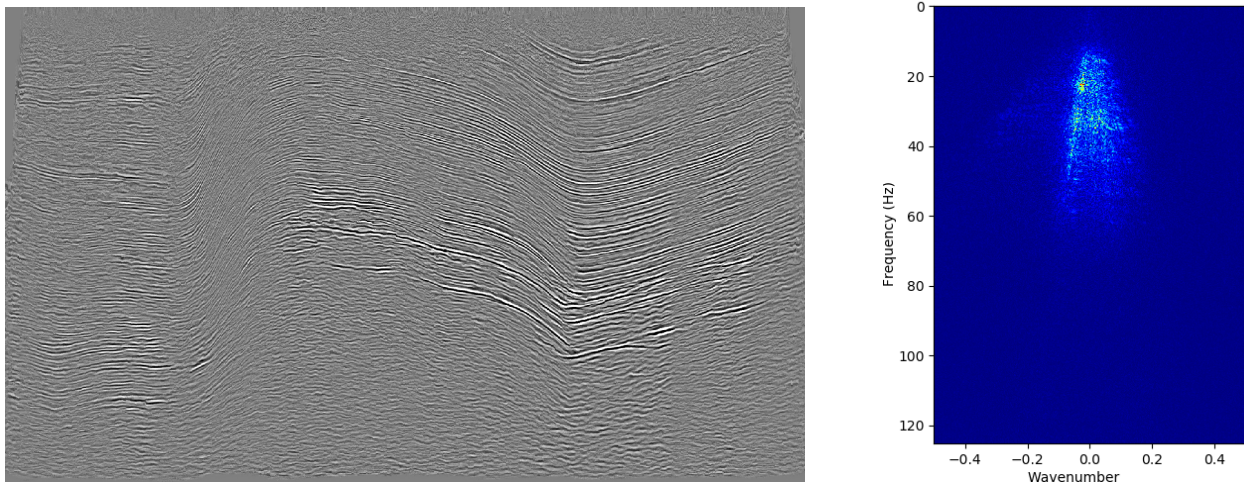
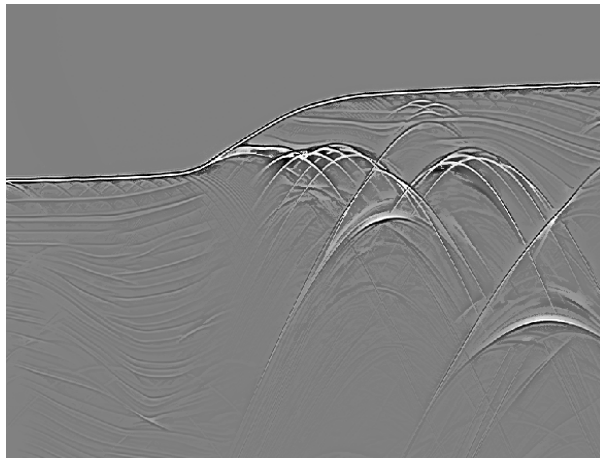
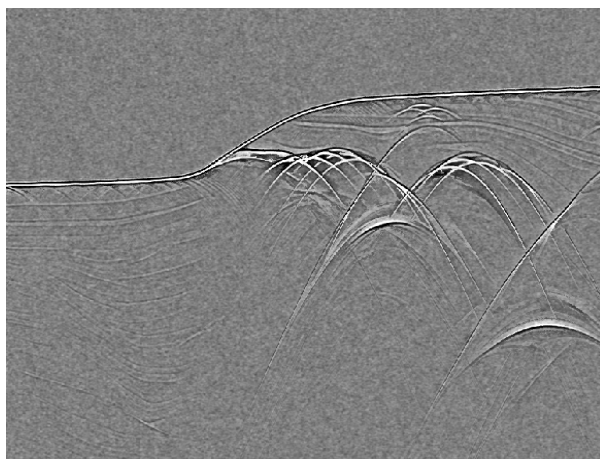


Figure 3: (left) Zero-offset stacked real reflection data Tacutu, used as style reference, and (right) its corresponding $f-k$ spectra.

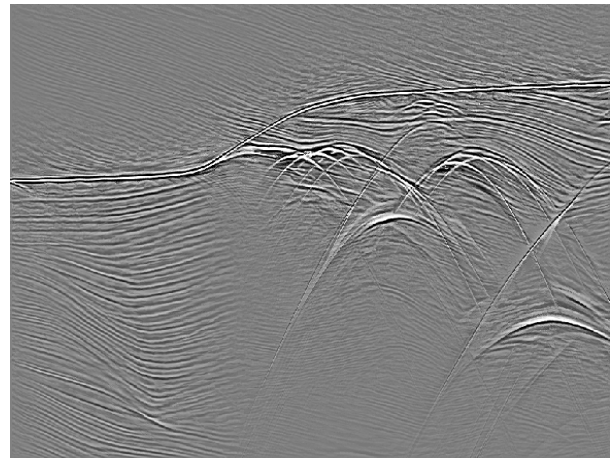


(a)

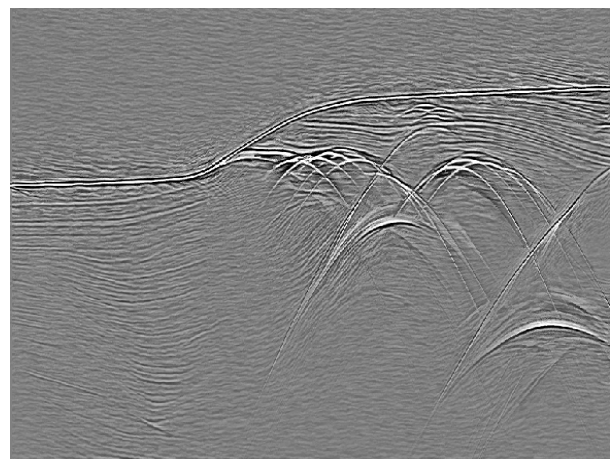


(b)

Figure 4: In (a) the synthetic reflection data Sigsbee. In (b) Sigsbee with added Gaussian noise, signal to noise rate set in 10.



(a)



(b)

Figure 5: Final outputs of the stylization process. In (a) the result of stylization process applied to the panel of Figure 4a. In (b) the stylization was made on the panel shown in Figure 4b.

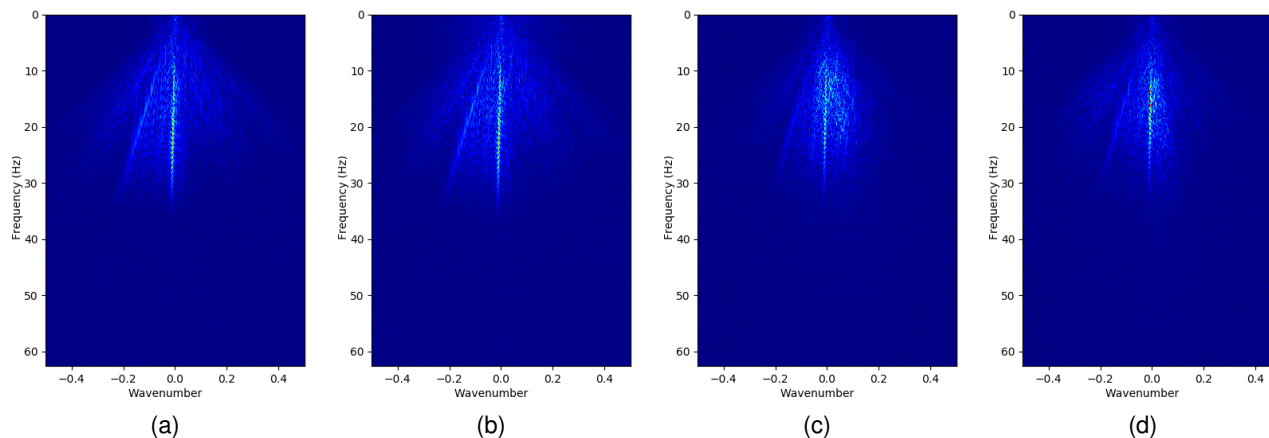


Figure 6: f - k spectra for each version of the Sigsbee data. (a) Sigsbee clean (Figure 4a). (b) Sigsbee with added Gaussian noise (Figure 4b). (c) Sigsbee clean stylized (Figure 5a). (d) Sigsbee noisy stylized (Figure 5b).

Figure 5. The NST technique has no concept of *ocean*, of course, so the water portion of the data would have to be muted back.

Figure 7 compares the same trace before and after the stylization process. We can clearly perceive the added noise, but peak location and zero-centeredness were preserved.

The pure Sigsbee's f - k spectra presents its events mainly concentrated in the negative neighborhood of the zero wavenumber (i.e. on the panel's left-hand side), while Tacutu has events on both sides. The f - k spectra of the data with zero-mean Gaussian, shown in Figure 6b, is highly similar to the pure Sigsbee's spectra Figure 6a, indicating that this type of noise does not add significant spectral content. In the examples enriched by NST (as shown in Figure 2), we see they incorporated both the real and synthetic data's signature. Therefore, in the Figures 6c and 6d, we see events occurring both in the positive and in negative wavenumber sides. This second approach significantly changes the original data spectral content.

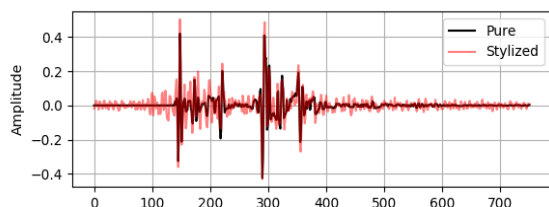


Figure 7: Comparison between the same trace in the stylized and original data. Peak location and zero-centeredness is preserved.

Limitations

The most evident drawback of the original Neural Style Transfer technique is the time and computational resources it demands. For each pair of data we need to combine, it is necessary to solve a large optimization problem. Although alternatives to the original procedures were developed in various works, as analyzed by Jing et al. (2017), many of them do not deliver good visual results to combinations of

arbitrary pairs of style and content.

Besides that, the effectiveness of a Neural Style Transfer technique depends on the problem's domain. For example, a photo-realistic style transfer requires special care with content preservation to avoid unwanted distortions. In our work, we solved this problem by simply adding back the original data to the stylized result, thus having some control over how much noise and distortion we want to insert in the synthetic data, but other alternatives merit investigation.

It is necessary to investigate further if some of the newer techniques available performs better in terms of the trade-off between time, computational resources and stylization quality. Also, we would like to quantitatively assess the impact of applying this technique to Machine Learning and Deep Learning seismic data experiments, evaluating its generality and robustness.

Conclusions

Neural Style Transfer is mainly associated with artistic works. In our study, we explored the use of the concept *style* to generate a synthetic data enriched with meaningful noise, combining the structural content of a mathematically modeled data with features from real data. We also discovered that inserting an initial random noise in the synthetic input yields better style transferring. These generated data could then be used to train more robust Machine Learning and Deep Learning systems, bypassing the scarcity of annotated data, or be used to test processing techniques such as migration or stacking.

Acknowledgments

We acknowledge support from the National Council for Scientific and Technological Development (CNPq-Brazil) and the Center for Computational Engineering and Sciences (FAPESP/CEPID #2013/08293-7-Brazil). S. Avila is partially funded by Google Research Awards for Latin America 2018 and FAPESP (#2017/16246-0). The authors also thank the High-Performance Geophysics (HPG) team for technical support. We also would like to thank Gatys et al. for letting their code available on GitHub.

References

- Araújo, L. M., F. M. C. Oliveira, J. H. Faccipieri, T. A. Coimbra, S. Avila, M. Tygel, and E. Borin, 2018, Detecção de estruturas em dados sísmicos com Deep Learning: Boletim SBGf. Publicação da Sociedade Brasileira de Geofísica, 18–21.
- Brenders, A., and J. Dellinger, 2016, Realistic signal-to-noise ratios for synthetic seismic data: Calibrating with measured noise and applications to waveform inversion, *in* SEG Technical Program Expanded Abstracts: Society of Exploration Geophysicists, 3992–3996.
- Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, 2009, ImageNet: A Large-Scale Hierarchical Image Database: Presented at the IEEE Conference on Computer Vision and Pattern Recognition.
- Faccipieri, J. H., T. A. Coimbra, L.-J. Gelius, and M. Tygel, 2016, Stacking apertures and estimation strategies for reflection and diffraction enhancement: *Geophysics*, **81**, V271–V282.
- Gatys, L. A., 2017, PytorchNeuralStyleTransfer. <https://github.com/leongatys/PytorchNeuralStyleTransfer>, accessed March 12th, 2019. (Implementation of the Neural Style Transfer algorithm in Pytorch).
- Gatys, L. A., A. S. Ecker, and M. Bethge, 2016, Image style transfer using convolutional neural networks: IEEE Conference on Computer Vision and Pattern Recognition, 2414–2423.
- Jing, Y., Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, 2017, Neural style transfer: A review: arXiv preprint arXiv:1705.04058.
- Lee, H., R. Grosse, R. Ranganath, and A. Y. Ng, 2009, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations: International Conference on Machine Learning, ACM, 609–616.
- Lu, P., M. Morris, S. Brazell, C. Comiskey, and Y. Xiao, 2018, Using generative adversarial networks to improve deep-learning fault interpretation networks: *The Leading Edge*, **37**, 578.
- Saxena, N., 2018, Seismic Soundoff, Episode 36. <https://seg.org/podcast/Post/6353/Episode-36-Image-processing-and-machine-learning>, accessed March 12th, 2019. (Podcast promoted by the Society of Exploration Geophysicists. Interview given to Andrew Geary).
- Simonyan, K., and A. Zisserman, 2014, Very deep convolutional networks for large-scale image recognition: arXiv preprint arXiv:1409.1556.
- Verma, P., and J. O. Smith, 2018, Neural style transfer for audio spectrograms: arXiv preprint arXiv:1801.01589.
- Wu, X., Y. Shi, S. Fomel, and L. Liang, 2018, Convolutional neural networks for fault interpretation in seismic images: Presented at the SEG Technical Program Expanded Abstracts.