Convolutional neural networks as aid in core lithofacies classification

Rafael Pires de Lima[1], Fnu Suriamin[2], Kurt J. Marfurt[3], and Matthew J. Pranter[3]

Abstract

Artificial intelligence methods have a very wide range of applications. From speech recognition to self-driving cars, the development of modern deep-learning architectures is helping researchers to achieve new levels of accuracy in different fields. Although deep convolutional neural networks (CNNs) (a kind of deep-learning technique) have reached or surpassed human-level performance in image recognition tasks, little has been done to transport this new image classification technology to geoscientific problems. We have developed what we believe to be the first use of CNNs to identify lithofacies in cores. We use highly accurate models (trained with millions of images) and transfer learning to classify images of cored carbonate rocks. We found that different modern CNN architectures can achieve high levels of lithologic image classification accuracy (approximately 90%) and can aid in the core description task. This core image classification technique has the potential to greatly standardize and accelerate the description process. We also provide the community with a new set of labeled data that can be used for further geologic/data science studies.

Introduction

Advances in deep learning and artificial intelligence promise to not only drive our cars but to also taste our beer (Gardner et al., 1994; Daily et al., 2017). Specifically, recent advances in the architecture of deep-learning convolutional neural networks (CNNs) have brought the field of image classification and computer vision to a new level. Very deep CNNs emerged in 2014 and have achieved new levels of accuracy in several artificial intelligence classification problems (Szegedy et al., 2014). The current benchmark in object category classification and detection, called ImageNet, consists of hundreds of mixed-object categories and millions of images (Deng et al., 2009; Russakovsky et al., 2015), and it is commonly used to train CNNs. Current CNN models are able to differentiate the image of a leopard from that of a container ship; moreover, they can differentiate images of leopards from their biological cousins — cheetahs and snow leopards (Krizhevsky et al., 2012).

Although machine learning has been significantly used in geoscience fields, the application of this technique in core-based lithofacies identification, a key component to better understanding oil and gas reservoirs, is still limited. Machine-learning techniques have been intensely used to aid seismic-facies classification (de Matos et al., 2007, 2011; Roy et al., 2014; Qi et al., 2016; Zhao et al., 2016, 2017; Qian et al., 2018), electrofacies classification (Allen and Pranter, 2016), lithofacies classification from well logs (Baldwin et al., 1990; Zhang et al., 1999; Bestagini et al., 2017), to predict permeability in tight sands (Zhang et al., 2018), and even for seismicity studies (Kortström et al., 2016; Perol et al., 2018; Sinha et al., 2018; Wu et al., 2018). Cored wells are important because they are the only data that provide the ground truth of subsurface reservoirs including the lithofacies variations. The goals of core-based rock-type descriptions are to identify key lithofacies and facies associations; evaluate facies stacking and identify and interpret depositional environments; evaluate the relationships among porosity, permeability, and lithofacies; and help operators to identify optimal zones for designing completions. Traditional core-based lithofacies identification is challenging because it is costly, time consuming, and subjective (e.g., different geologists describing the same core might yield different results). To address some of the core-based lithofacies identification challenges, we evaluate whether a CNN can help a specialist on their image-recognition task.

CNN goes hand in hand with the construction and archival of digital databases. Many museums are now

[1]The University of Oklahoma, School of Geology and Geophysics, 100 East Boyd Street, RM 710, Norman, Oklahoma 73019, USA and The Geological Survey of Brazil-CPRM, 55 Rua Costa, São Paulo, Brazil. E-mail: rlima@ou.edu (corresponding author).

[2]The University of Oklahoma, School of Geology and Geophysics, 100 East Boyd Street, RM 710, Norman, Oklahoma 73019, USA and Oklahoma Geological Survey, 100 East Boyd Street, Room N-131, Norman, Oklahoma 73019, USA. E-mail: huangcienming-1@ou.edu.

[3]The University of Oklahoma, School of Geology and Geophysics, 100 East Boyd Street, RM 710, Norman, Oklahoma 73019, USA. E-mail: kmarfurt@ou.edu; matthew.pranter@ou.edu.

busy digitizing and sharing their collections (Blago-derov et al., 2012; Ellwood et al., 2015) With the exception of core measured by deep-sea drilling projects and the like (e.g., NOAA, 2016), core images are not readily available. As an example, more than 100 mi of cores are stored in the Oklahoma Petroleum Information Center, managed by the Oklahoma Geologic Survey. Other states and countries have similar repositories (USGS Core Research Center, 2018). Further digitization of this valuable resource resulting in core images will not only facilitate access to data for traditional analysis but will also provide the information needed to build and calibrate innovative machine-learning algorithms. The work we use here has the potential to organize many miles of slabbed cores into a reliable and coherent system easily accessible to a variety of users.

In this paper, we provide one of the first attempts to conduct automated core lithofacies classification using CNN. We begin with an overview of the methodology, which includes data preparation and transfer learning. The details of the CNN method are summarized in tutorial form in Appendix A. Then, we apply CNN to our core data set, and we use confusion matrices, test and validation accuracies, as well as precision, recall, and the F1 score (Fawcett, 2006) computed with the final test set as a means to analyze our results. We conclude with a summary of our findings and suggestions on how our workflow can be extended and improved.

Methodology

The deep-learning methodology and CNN techniques are now very well-disseminated in diverse fields. LeCun et al. (2015) present details in the construction and the value of deep learning. Dumoulin and Visin (2016) give details on convolutions and other arithmetic steps used in deep-learning algorithms. Although carefully constructed interative papers have been published detailing CNN image transformations and image understanding (e.g., Olah et al., 2017, 2018), CNN may appear to be "magic" and therefore somewhat suspect to the practicing geoscientist. For this reason, Appendix A provides a tutorial that looks under the covers, providing a simple CNN application to classify images into three groups. The work for this paper was developed using open-source computational packages described by Hunter (2007), Chollet (2015), and Abadi et al. (2016)

When used for image recognition tasks, CNN models need examples (images) to understand the properties of each "class" that they try to discriminate. Part of the parameters learned for a primary task (such as the ImageNet classification) can be transferred to a secondary task (e.g., lithofacies classification) through the use of transfer learning (Pan and Yang, 2010; Oquab et al., 2014; Yosinski et al., 2014). Our work focuses on using transfer learning of complex CNN architectures to serve our specific image recognition task. The following subsections detail how we prepared our data sets and give a brief explanation of transfer learning.

Data preparation

We used cores described using traditional methods published by Suriamin and Pranter (2018), capturing images using modern photographic equipment to generate the set of labeled data to feed our CNN. The total section used for this project consists of approximately 700 ft from one core from the Mississippian limestone and chert reservoirs in the Anadarko Shelf, Grant County, Oklahoma. The set of core images shown in Table 1 includes 17 different lithofacies. Two pairs of lithofacies exhibit similar lithology and appearance; we grouped these into a single class for this project. We carefully cropped the images in a standardized fashion, providing consistent input to the CNN. We used a sliding window technique to extract consistent squared cropped sections from the original core images (Figure 1), generating 180×180 pixels images representing

Table 1. Class number assigned to each lithofacies in the core used in this study.

Class	Lithofacies	Training set	Test set
01	Chert breccia in greenish shale matrix	218[*]	3
02	Chert breccia	236[*]	3
03	Skeletal mudstone-wackestone	258[*]	4
04	Skeletal grainstone	160[*]	3
05	Splotchy packstone grainstone	344[*]	4
06	Bedded skeletal peloidal packstone-grainstone	416[*]	4
07	Nodular packstone-grainstone	445	11
08	Skeletal peloidal packstone-grainstone	Not used	Not used
09	Bioturbated skeletal peloidal packstone-grainstone	795	19
10	Bioturbated mudstone-wackestone	150[*]	4
11	Brecciated spiculitic mudstone	Not used	Not used
12	Intraclast spiculitic mudstone	Not used	Not used
13	**Spiculitic mudstone-wackestone**	**3077**	**79**
14	**Argillaceous spiculitic mudstone-wackestone**		
15	Glauconitic sandstone	Not used	Not used
16	**Shale**	**789**	**17**
17	**Shaly claystone**		
Total number of images in each set		6888	151

Note: Classes 13-14 and 16-17 in bold exhibited a similar lithology and appearance so are combined into two classes instead of four. During training, the training set data are further split: 10% are randomly selected to be part of a validation set, and 5% are randomly assigned as a training test set. The proportion used for validation and test splitting is commonly dependent on the number of samples available and the type of machine-learning model being trained. CNN models usually improve with more examples; therefore, we selected a smaller percentage to be part of the validation and test sets. The last column of this table (the test set) comprises the selected images described in Figure 1, and it is the test set used for further analysis in this paper. Classes with fewer than 30 original images were not used in this study (modified from Suriamin and Pranter, 2018).

*Classes that were augmented by horizontally flipping the images.

roughly 2×2 in of cored rock. Note in Figure 1 that the heavily damaged rock is not present in the images used for training/testing. We chose to eliminate these images because they would increase variability within the class. Ideally, more core data would provide sufficient images to define damaged classes. The sliding window cropping process augments the number of images of our initially small collection, thereby further generalizing the CNN. Some classes contained less than 300 images. To augment the representation of those classes, we doubled the number of input images by flipping the image horizontally. Then, we select approximately 2% of the original data for each class to serve as the test data. During training, 5% of the total training data are randomly selected to be part of the training test. The training test set is used for an overall performance evaluation. We provide more detailed analysis using the test set. The selection of images to be part of the test has a higher standard than the images selected to be part of the training test. Because each image selected to be part of the test set forces us to discard its neighbors (Figure 1), we select only 2% of the original data to be part of the test set.

Even after image augmentation, Table 1 shows that some classes have a significantly larger number of images than others. This difference in amount of labeled data for different classes is referred to as class imbalance (Japkowicz and Stephen, 2002; Buda et al., 2018) and can cause undesirable effects when training classifiers. In this study, we did not notice a significant bias caused by such class imbalance. Therefore, although we did not augment it, we chose to retain all of the images in the most common 13-14 Spiculitic mudstone-wackestone class. In contrast, we removed from analysis classes represented by less than 30 images in which the initial testing indicated that these undersampled classes were reducing CNN accuracy.

Transfer learning

Transfer learning is a powerful technique that can be used to address the shortage of sufficient domain-specific training data (Carranza-Rojas et al., 2017). In transfer learning, the learned parameters of a base model trained on a base data set are applied to a different task (Yosinski et al., 2014). In our application, we use a CNN model trained to identify the images of the ImageNet challenge to classify lithofacies in core (Figure 2). ImageNet is a data set consisting of thousands of classes ranging from biological and household images to vehicles and bridges; to our knowledge, no rock or core images were included in its construction. Another advantage of using transfer learning is

to reduce the training computation time by using the trained layers as feature extractors (Appendix A) and rather training only a new classification layer. Examples of transfer learning include Carranza-Rojas et al. (2017) for herbarium specimens, Esteva et al. (2017) for skin cancer classification, and Gomez Villa et al. (2017) for camera-trap images. Tajbakhsh et al. (2016) use different medical imaging applications and perform a comparison between CNNs trained from scratch with the pretrained CNNs. The authors found that using a pretrained CNN frequently outperforms a CNN model trained from scratch especially when limited training data are available.

When CNNs are trained with natural images, the first layers of the deep neural network learn features that are useful to identify textures or colors. This behavior is quite common in CNN models; the analysis is reevaluated if the initial layers learn image properties other than color or texture. Because of this CNN characteristic, models with good performance trained on the ImageNet challenge (e.g., Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2014, 2015; He et al., 2016; Zoph and Le, 2016; Zoph et al., 2017; Sandler et al., 2018) can be successfully retrained for new, field-specific classification problems (e.g., Tajbakhsh et al., 2016; Carranza-Rojas et al., 2017; Esteva et al., 2017; Gomez Villa et al., 2017; Norouzzadeh et al., 2018).

In this project, we evaluate transfer learning using four different trained models: InceptionV3 (Szegedy et al., 2015) consisting of 48 layers, ResNetV2 — implemented with 50 layers (He et al., 2016), MobileNetV2 (Sandler et al., 2018) with 20 layers, and NASNet (Zoph
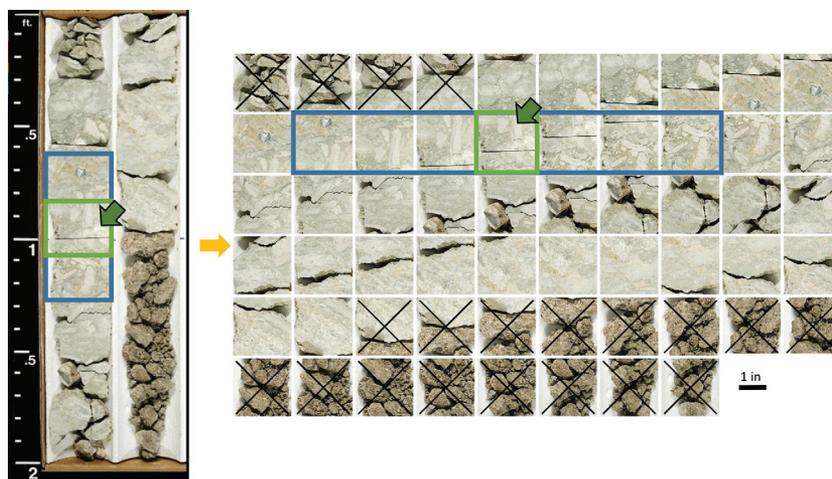

Figure 1. The image augmentation of a photographed core, the core using a sliding window of cropped image. This approach provides the CNN with a greater amount of training data. The blue rectangle shows images that were never used during training (the test data). The cropped images crossed were discarded from the data sets (damaged rocks). The green arrow indicates a random image that could have been selected to be part of the test set. When an image like this is selected, the overlapping neighboring images are also removed from the training set. The separation of test data was the same for all classes in this project.

and Le, 2016; Zoph et al., 2017) with 20 layers. These models and the learned parameters are publicly available and can be downloaded from TensorFlow Hub (2018) website. Each one of the CNN models requires different-sized images as input: 299×299 pixels for InceptionV3, 224×224 pixels for ResNetV2, 224×224 pixels for MobileNetV2, and 331×331 pixels for NAS-Net. Because our images are 180×180 pixels in size, we use simple bilinear interpolation to conform to the size of the transfer learning model used. As described in Appendix A, all subsequent layers are dependent on the size of the input data. Appendix A shows how transfer learning is achieved by using the convolutional layers as feature extractors for our core images thereby facilitating the training of a densely connected classification layer.

Carranza-Rojas et al. (2017), Esteva et al. (2017), and Gomez Villa et al. (2017) each use some 100,000 images in their data sets to perform transfer learning. Although we have a significantly smaller data set consisting of less than 7000 images, we still achieved a high level of accuracy as presented in the next section. We use confusion matrices, test and validation accuracy, and precision, recall, and F1 score (Fawcett, 2006) computed with the final test set as a means to analyze our results.
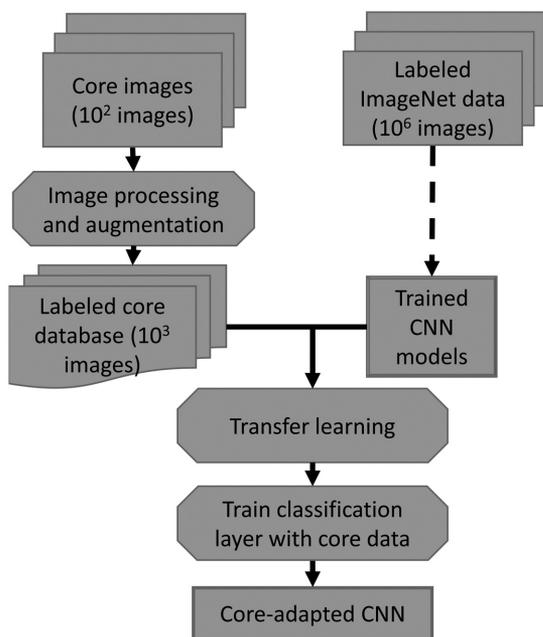

Figure 2. Flowchart summarizing the workflow used in this paper. We begin with photographic images of the slabbed core, followed by simple image processing and data augmentation to generate our core image database. The CNN models that we use as feature extractors were previously trained using millions of images on the ImageNet challenge. We then use transfer learning and reuse the ImageNet data set classification CNN weights. Finally, we train the last layer to provide the desired core image classification.

Results

In this section, we present the overall results that we obtained as well as examples of the classifications performed by the retrained CNNs. Because the results of the four chosen CNNs are similar, in this section we show details of the ResNetV2 retrained CNN (apart from the training test set in Table 2). Plots and tables regarding the other three CNNs are presented in Appendix B.

The results for the training test data set are showed in Table 2.

These training tests accuracies were achieved after 5000 iterations using a gradient descent algorithm. Figure 3 shows the training and validation accuracy result for each step of the gradient descent. The CNN quickly reaches satisfactory levels of accuracy. After performing feature extraction, only the last classification layer needs to be trained. The training time in a single-core CPU with 3.60 GHz clock speed does not exceed 1 h for our data set for any of the four CNN models used. Access to graphical processing units provides even greater computation speeds.

Table 2. Training test set data results for the different models used for transfer learning.

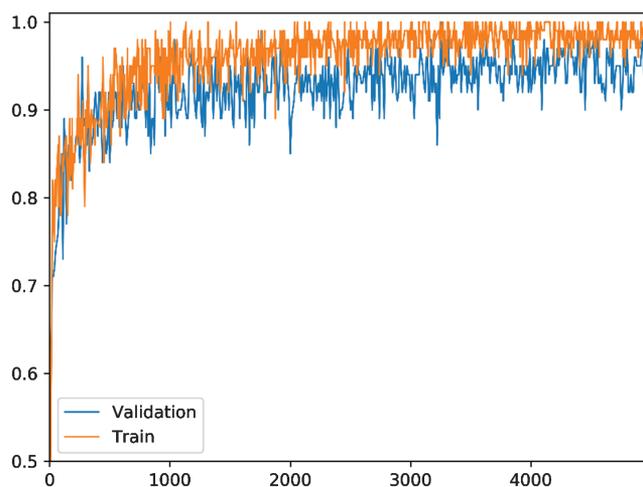| Model	Training test accuracy
InceptionV3	0.95
ResNetV2	0.95
MobileNetV2	0.95
NASNet	0.90

Figure 3. Validation and training accuracy for the ResNetV2 training. Note that after approximately 1000 iterations, the gains are marginal. Because the cost of training the classification layer is inexpensive compared with training the entire model, we can afford to let the model train for many steps.

Figure 4 shows representative images from the test data set classified using the retrained ResNetV2. Because the CNN provides different levels of probability

Table 3. Precision, recall, F1 score, and support for the classification performed by the retrained ResNetV2.

Class	Precision	Recall	F1 score	Support
01	1.00	1.00	1.00	3
02	1.00	0.33	0.50	3
03	0.80	1.00	0.89	4
04	1.00	1.00	1.00	3
05	0.67	1.00	0.80	4
06	0.75	0.75	0.75	4
07	0.89	0.73	0.80	11
09	0.90	0.95	0.92	19
10	0.57	1.00	0.73	4
13-14	0.99	0.95	0.97	79
16-17	0.94	0.94	0.94	17
Weighted	0.93	0.92	0.92	

Note: The last row shows the weighted values for each one of the metrics.

when assigning the classes, we can define an acceptable threshold to accept a given prediction. Choosing different values for this threshold value is also a commonly used tool to analyze the performance of a classifying algorithm. Ferri et al. (2003), Everson and Fieldsend (2006), and Fawcett (2006) give details of the receiver operating characteristics graphs that arise when performing such an analysis. In this paper, we choose the threshold to be 0.30; this means that we accept the image classification given by the CNN when any possible class receives a probability higher than 0.30. This value was chosen so that all images would be classified, even if the CNN is not very confident. Such a threshold value is enough for our model to assign a class for each one of the images in the test set. Figure 5 shows the confusion matrix generated when the test set is classified by the retrained ResNetV2. Precision, recall, F1 score, and support as well as weighted precision, recall, and F1 score are presented in Table 3. All these metrics range from 0 (poor performance) to 1 (good performance). Precision and recall indicate how often the model was correct predicting the analyzed class. Precision is defined as the ratio of true positives and the sum of true positives and false positives. Recall is defined as the ratio between true positives and the sum of true positives and false negatives. F1 is the harmonic average of precision and recall.
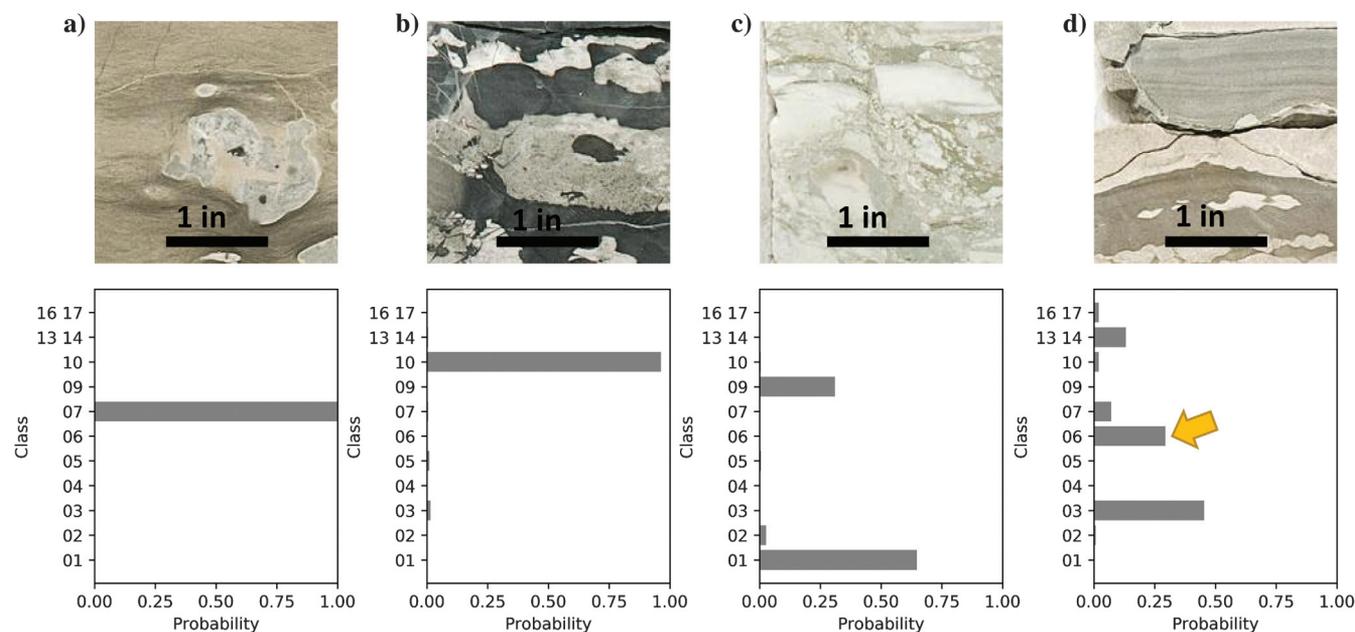

Figure 4. Examples of the classification performed by the retrained ResNetV2. (a) The CNN very confidently assigned the image to the correct class (class 07, Nodular packstone-grainstone). (b) Again, the CNN provides a high level of confidence to assign the image to the correct class (class 10, bioturbated mudstone-wackestone). (c) The CNN still assigns the image to the correct class, but with lower confidence (class 01, chert breccia in the greenish shale matrix is the correct class). (d) The image shows an example in which the CNN failed to correctly assign the class. The CNN assigned a higher confidence for class 03 (skeletal mudstone-wackestone, with 0.45 probability), whereas the correct class is actually class 06 (bedded skeletal peloidal packstone-grainstone, 0.29 probability, the yellow arrow in the image). Setting a confidence threshold of 0.50 or greater would identify this classification as "ambiguous," calling for human intervention.

Discussion

To our knowledge, this is the first study conducted using slabbed core image classification using CNNs. With comparable metrics for all the CNN architectures tested, we observe a high level of concordance (and confidence) between the expert labeled data and the classifications suggested by the CNNs. Using the methodology that we presented in this paper, a user can obtain the probability that a standardized picture of a core belongs to one of the described lithofacies even if the user has little experience with the core description. This capability can not only accelerate the interpretation of large data volumes by using nonexpert technologists, but it can also identify inconsistencies in the interpretation between different experts working on the same data. This potential inconsistency suggests that we construct a human interpreter confusion matrix comparing multiple interpretations of the same core. This confusion matrix can then be constructed from a single interpreter and the CNN. Identification of such inconsistencies within teams composed of members with different backgrounds promises to facilitate data comprehension and accelerate project advancement. Even though we use a relatively small database of images, Figure 5 and Table 3 show that the retrained ResNetV2 achieved high levels of accuracy. The remaining three architectures' (InceptionV3, MobileNetV2, and NASNet) results also show high levels of accuracy (Appendix B).

When performing a core description, a human interpreter relies on texture, structures, and pattern analysis to define the lithofacies being analyzed. In this manner, the classification performed by CNNs somewhat mimics human classification. Nonetheless, when a geologist is describing a rock, other rock properties (not visual properties) can be analyzed by the interpreter. Does the mineral react with acid? How hard is the mineral? Therefore, when using CNN models, the user needs to remember that the best result that the CNN can provide is only the best result achieved by a visual (and strictly nontactile) analysis of an image. We can, however, modify the deep-learning architectures to be multidimensional. Especially, the digital images can be augmented by measures of resistivity, density, X-ray fluorescence, Fourier transform infrared spectroscopy, and other measures to produce an even more powerful tool.

In the architecture that we used here, any image used as an input to the CNN classifier will predict that the image belongs to one or more of the CNN's learned classes. This means that the CNN will never declare the image to be none of the predefined classes. Figure 6 shows an image of a carpet classified by the retrained ResNetV2.

Although there are visual similarities between the carpet and the images in the training set, the resulting classification demonstrates the necessity of quality controlling CNN output.

Because the CNN models are trained with expert labeled data, such expertise is abstractly maintained in the different parameters optimized in the CNN. Consequently, we can absorb interpretations performed by different specialists and save them in unique CNN models. Such a data capture would provide a way of sharing geologic knowledge across great distances.
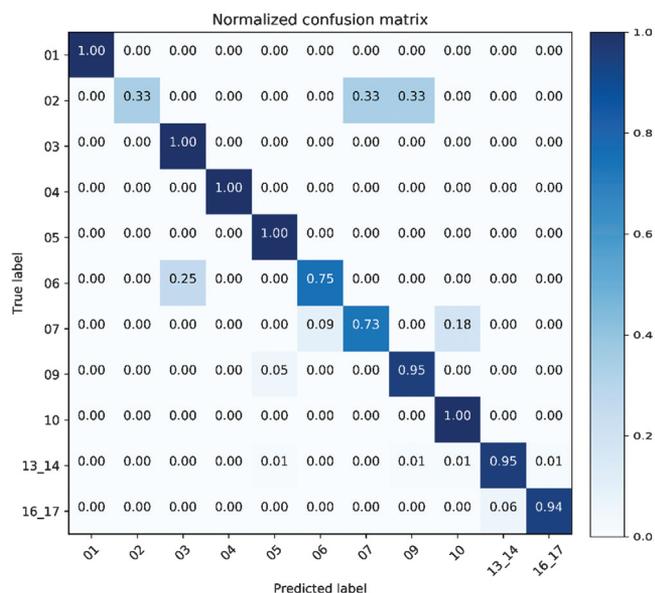

Figure 5. Normalized confusion matrix of the retrained ResNetV2 applied to the test set. Refer to Table 1 for the class lithofacies and the number of images for each class.
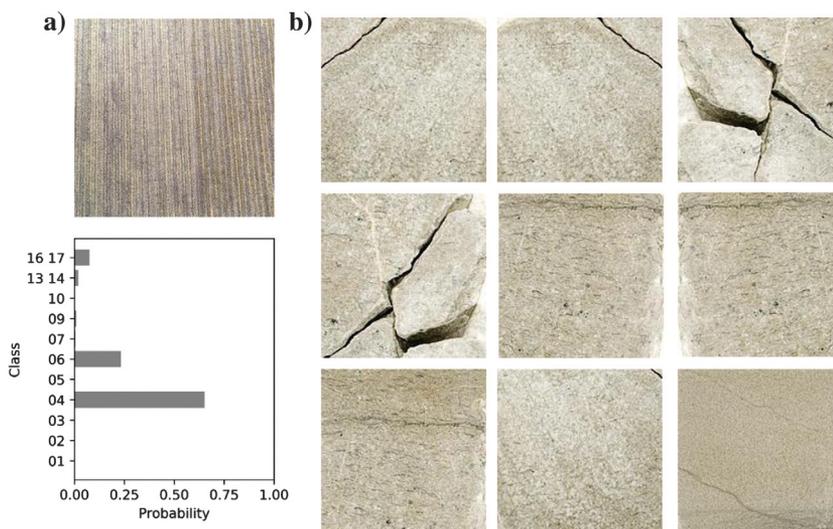

Figure 6. (a) A photographic image of a carpet classified by the ResNetV2 and (b) examples of images from the class 4 training data set. The CNN is 70% confident that the carpet belongs to class 4 — Skeletal grainstone.

Geoscientist working in challenging (or unfamiliar) geologic settings can access the knowledge of a wide range of experts through the use of properly trained CNNs.

Human geoscientists will not be replaced by machine learning. Clearly, expert geoscientists are required to construct the labeled training data. Expert scientists are also required to quality control the prediction, perhaps manually examining all predictions that exhibit less than a threshold of confidence. Emulating the approach of human geoscientists who subliminally apply models of deposition and diagenesis while examining the core will be very difficult. Linking different lithologies within a parasequence is a part of human interpretation. At the present, computers have a difficult time with such geoscientific image segmentation problems.

Suggestions for further study

During the course of the work, we had access to pictures of a single core, and we presented the results that we obtained using modern CNN models for classification for that single interpreted core. We envision that the process used in this paper can be used to greatly accelerate the interpretation of multiple cores. Users can achieve such multicore interpretation result with a more iterative approach: An experienced expert labels the key lithofacies of the region; the CNN is then trained and classifies the remaining cores. The results of such classification are then evaluated by the expert — a form of active learning (Settles, 2012; Sener and Savarese, 2018). If necessary, the user can retrain the CNN with a now-increased set of labeled images (the originally expert-labeled images and the new CNN-labeled images). In this manner, many miles of core can be interpreted with lower effort. Several new challenges can arise when working with historical data of lesser quality, different formats, with different interpretations, and from different well locations sampling different geology. This paper shows one successful application of a growing technology; however, different evaluations need to be addressed for every specific task. When these extra variables exist, such as poor-quality data or multiple wells with inconsistent interpretations, it is likely that the performance will be negatively affected.

For this project, we relied on standardized core pictures and a simple sliding window to extract images. Therefore, it is reasonable to assume that some images will show more than one lithofacies. Different ways of data acquisition can further improve the results of CNN models.

Conclusion

In this paper, we provide one of the first attempts to conduct automated core lithofacies classification using CNNs. The methodology we use does not depend on specialized bench work and can be applied to existing images of slabbed cores.

Efforts in data digitization are important initiatives to preserve scientific knowledge, and the approach we use here can be improved with information generated from such endeavors. The development of customized core databases can be of extreme value for companies and researchers that have work that is dependent on core descriptions. When operators need to reevaluate prospective plays — due to new acreage acquisition or to update the geologic knowledge with modern geologic information — thousands of feet of expensive slabbed core might be overlooked due to time and personnel constraints. Further development of the project that we present here will ultimately speed up the process of core description with the use of slabbed core-specific CNN models.

Using the technology that we applied in this paper, an experienced geologist describes a small percentage of the core using traditional, careful, and standardized "visual and tactile" lithologic description and then uses that information to train a CNN. The trained CNN can then classify the remaining core — the geologist can quality control the results and will have more time to work on the necessary details. The methodology that we used here can be used to standardize interpretation in large collections of core data. Because interpretation might be subjective, teams can choose to maintain "the best" interpreter knowledge abstractly captured by a CNN trained with data labeled by the most experienced geologist or to train a CNN using only new concepts. A task that would be infeasible now (having different specialists interpreting miles of core) can be achieved with the help of CNN models.

Acknowledgments

Data and materials availability

Data associated with this research are available and can be obtained by contacting the corresponding author.

Appendix A
CNNs intuitions

Although deep learning and CNNs seem to have become buzzwords, the intuition of how these techniques

work might be obscured for some of us. Although CNN models are becoming increasingly complex, the building blocks are very familiar to geoscientists. Convolutions — performed in one, two, three, or n-dimensions — are the same operations we become familiarized when dealing with a seismic wavelet convolving with the earth's reflectivity series. Many seismic attributes are also based on convolutions. Many apps and software offer the user the option to extract the edges of an image — or to blur such an image. These are just a few of many convolution operations that we commonly encounter. Our objective with this appendix is to give a short and informal overview of the essentials of CNNs.

When using CNNs for image classification tasks, the models use the resulted filtered image (an image convolved with a convolution kernel) as input to another operation (or the next layer). The deep-learning nomenclature comes from this pattern — the input of a layer is used as the input to the next one.

In this appendix, we have an easy task for a CNN: to classify three classes of very distinct images (Figure A-1). Each class has 10 examples of RGB images 180×180 pixels that were extracted from pictures of a slabbed core in the Mississippian limestone and chert reservoirs. Geologists can easily tell the difference between the classes and could probably correctly name the lithofacies even with these low-resolution images. The CNN actually needs the interpreter (the domain expert) to correctly label the images and separate them — in

our simple example we are calling the lithofacies class 1, class 2, and class 3.

We design a not-so-deep CNN shown in Figure A-2. This CNN is composed of six layers: convolution, max
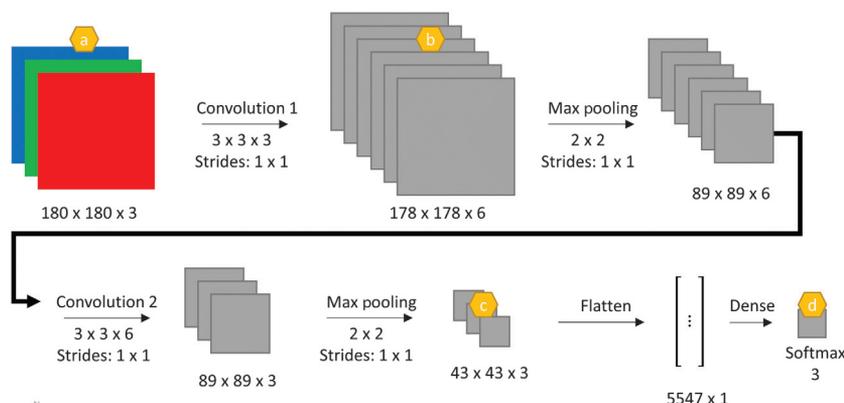

Figure A-2. A simple CNN. The golden hexagons show images displayed in the next figure. In this toy example, a set of images with a size of 180×180 pixels is input to a CNN with six layers. The first layer is a set of six convolution kernels with a size of $3 \times 3 \times 3$. The value of the third dimension is the same as the value of the number of channels of the previous layer. Note that after the first convolution, the object reduces in height and width, but its number of channels increases. For the next step, a max pooling (an operation in which we extract the maximum value of a submatrix of the input) further reduces the height and width. This "thinner" object is then input to another convolution layer following by another max pooling. After the last max pooling, the layer is then flattened, meaning that all its values are stored as a single vector. The last layer uses as input all the values of the flattened vector to compute the probability that the input image belongs to one of three classes. Note that with this architecture, whatever is used as input will output some probability of belonging to one of the three classes. The kernels of the convolutional layers and the softmax of the last layer are the parameters that need to be trained for this neural network. In this example, we need to train a total of 16,977 parameters. For convolution 1, we need to train $3 \times 3 \times 3 \times 6 + 6$ (bias) = 168 parameters and for convolution 2, 165 parameters. The dense layer is responsible for 16,644 parameters that need training. These ratios (convolution and dense parameters) should not be used as a comparison with more complex CNNs such as the one we used in the main body of this study; truly deep CNN will have many more convolution parameters to be optimized.
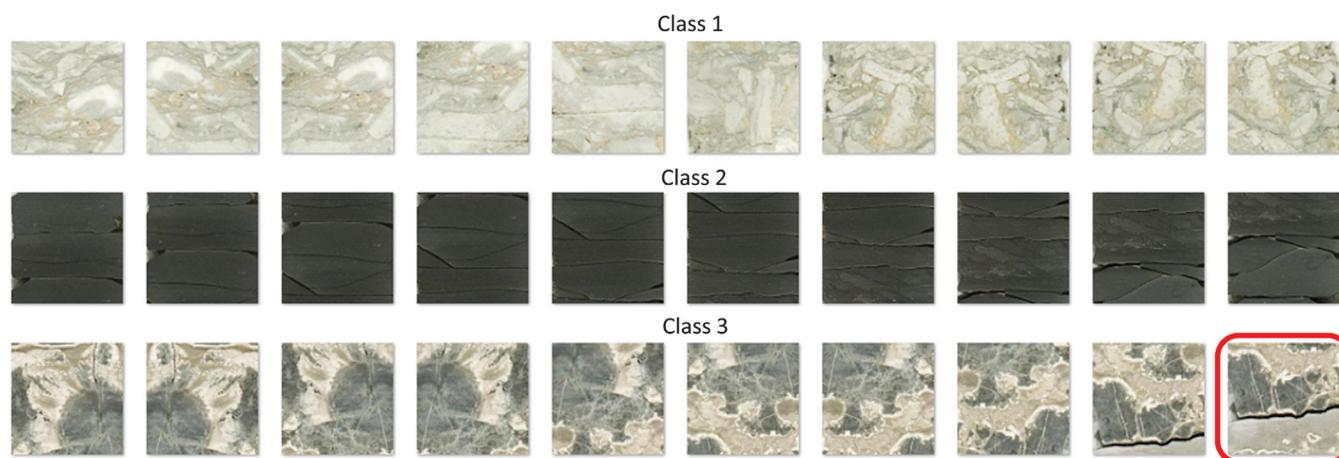

Figure A-1. Our very simple set of images used in a toy CNN example. We highlighted the image in class 3 that is used as an example in Figure A-3. Because all images are very similar in their set, this is an easy task for CNN models and we can achieve high accuracy with a simple network.

pooling, convolution, max pooling, flatten, and dense layers. The pooling layer extracts statistics of submatrices of the input data. In this case, we are using maximum (max); therefore, these operations look at a submatrix of the input and keep the maximum value of that submatrix to be the input for the next layer. The flatten layer restructures the data to be a single column vector. The dense layer is the traditional neural network composed of a linear transformation followed
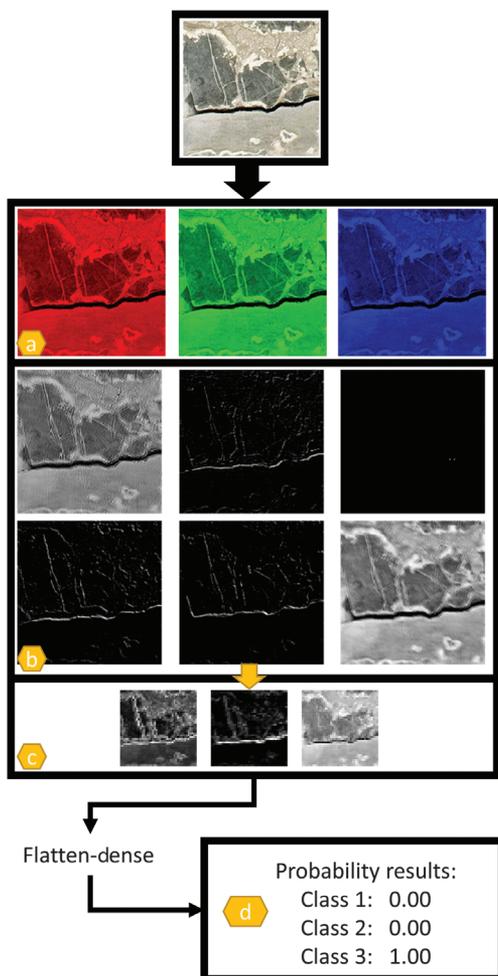
by a nonlinear transformation (softmax in this case) that are densely connected, i.e., each one of the elements is connected to each one of the neurons in the upcoming neuron. The output of the dense layer is the probability that an image belongs to classes 1, 2, or 3.

During training, all of the randomly started parameters are optimized to reduce the cost function. The cost function is commonly defined as the sum of the loss/error of an image being assigned in the wrong class in the training set. This simple example has only a training set; we did not set the validation and test set as would be appropriate for a real machine-learning methodology. Therefore, after training, we have this set of parameters that can take an image, perform different operations on that image, and come up with a value of how probable the image belongs to one of the training classes. This image transformation is displayed in Figure A-3. Note that the CNN very confidently sets the image as belonging to class 3 (with 1.00 confidence). This result can be achieved because the classes are very well-defined, and the images have low variance (all images in the same class are very similar to each other). As can be seen in Figure A-3, this small CNN trained convolutional kernels that are very good to detect edges. If we wanted to use transfer learning with this CNN, we would "delete" the last layer (softmax, gold hexagon d) and add a new classification layer.

Figure A-3. Simplified workflow and the resulting images extracted from different layers when the figure on the top left is input to the CNN shown in Figure A-2 after training. The golden hexagons can be used for easier reference between this figure and Figure A-2. Note how the set of weights (the convolutional kernels) learned by this CNN learns how to identify edges in the input image. This is a common behavior in CNNs when used with natural images (Yosinski et al., 2014). In a sense, much as a trained geologist, the CNN learns how to identify different patterns. Note that the image in frame a is a simple decomposition of the original image; therefore, we choose to display them as the red-green-blue color. The images in frames b and c are results of different "filters" applied in different steps of the CNN and are composed of a single channel (the convolution kernels have different sizes as shown in Figure A-2). We choose to display these images in grayscale.

Appendix B

InceptionV3, MobileNetV2, and NASNet metrics

In this appendix, we show the metrics for the retrained CNN models not presented in the main text.

Table B-1. Precision, recall, F1 score, and support for the classification performed by the retrained InceptionV3.

Class	Precision	Recall	F1 score	Support
01	0.75	1.00	0.86	3
02	0.33	0.33	0.33	3
03	0.50	0.75	0.60	4
04	0.67	0.67	0.67	3
05	0.60	0.75	0.67	4
06	0.50	0.75	0.60	4
07	0.82	0.82	0.82	11
09	0.75	0.79	0.77	19
10	0.80	1.00	0.89	4
13-14	0.96	0.86	0.91	79
16-17	0.88	0.88	0.88	17
Weighted	0.85	0.83	0.84	

Note: The last row shows the weighted values for each one of the metrics.

InceptionV3

Training and validation accuracy results for each step of the gradient descent are presented in Figure B-1. Figure B-2 shows the confusion matrix generated when the test set is classified by the retrained InceptionV3. Precision, recall, F1 score, and support as well as weighted precision, recall, and F1 score are presented in Table B-1.

MobileNetV2

Training and validation accuracy results for each step of the gradient descent are presented in Figure B-3. Figure B-4 shows the confusion matrix

Table B-2. Precision, recall, F1 score, and support for the classification performed by the retrained MobileNetV2.

Class	Precision	Recall	F1 score	Support
01	1.00	0.67	0.80	3
02	0.40	0.67	0.50	3
03	0.75	0.75	0.75	4
04	0.67	0.67	0.67	3
05	0.75	0.75	0.75	4
06	0.57	1.00	0.73	4
07	0.90	0.82	0.86	11
09	0.75	0.79	0.77	19
10	0.80	1.00	0.89	4
13-14	0.96	0.91	0.94	79
16-17	0.94	0.88	0.91	17
Weighted	0.89	0.87	0.87	

Note: The last row shows the weighted values for each one of the metrics.

Table B-3. Precision, recall, F1 score, and support for the classification performed by the retrained NASNet.

Class	Precision	Recall	F1 score	Support
01	0.75	1.00	0.86	3
02	0.67	0.67	0.67	3
03	0.50	0.75	0.60	4
04	0.60	1.00	0.75	3
05	0.43	0.75	0.55	4
06	0.50	0.75	0.60	4
07	0.69	0.82	0.75	11
09	0.77	0.68	0.72	19
10	0.80	1.00	0.89	4
13-14	0.93	0.81	0.87	79
16-17	0.81	0.77	0.79	17
weighted	0.82	0.80	0.80	

Note: The last row shows the weighted values for each one of the metrics.

generated when the test set is classified by the retrained MobileNetV2. Precision, recall, F1 score, and support as well as weighted precision, recall, and the F1 score are presented in Table B-2.

NASNet

Training and validation accuracy results for each step of the gradient descent are presented in
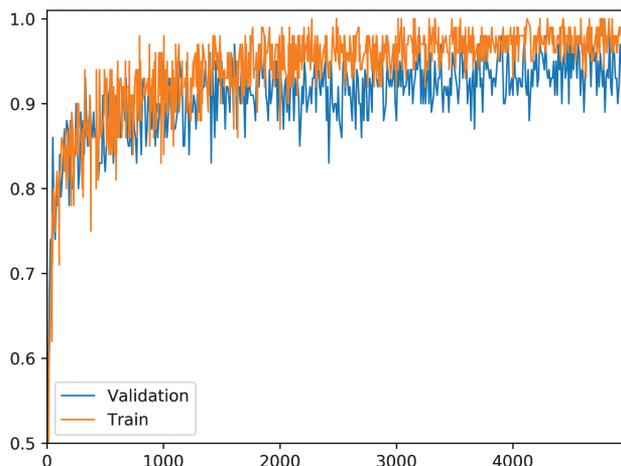

Figure B-1. Validation and training accuracy for the InceptionV3 training. Note that after approximately 2000 iterations, the gains are marginal. Because the cost of training the classification layer is inexpensive, we can afford to let the model train for many steps.
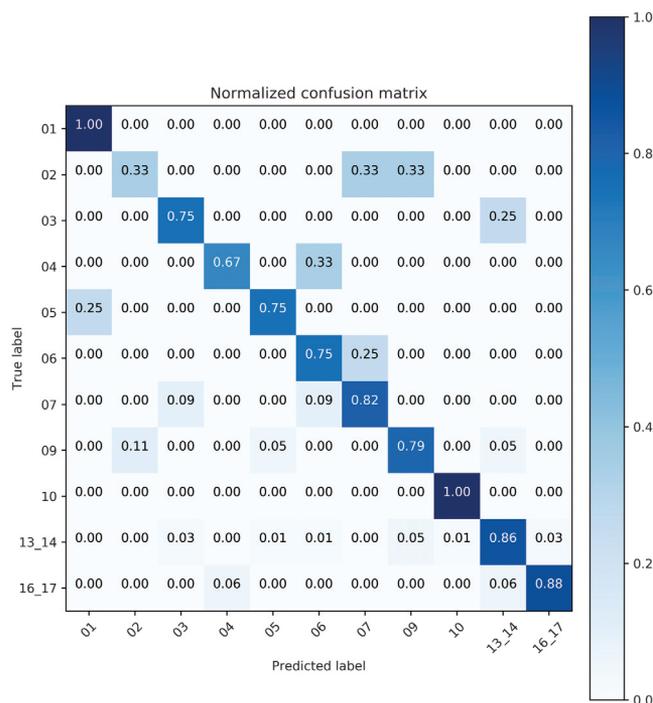

Figure B-2. Normalized confusion matrix of the retrained InceptionV3 applied to the test set. Refer to Table 1 in the main text for class lithofacies and the number of images for each class.

Figure B-5. Figure B-6 shows the confusion matrix generated when the test set is classified by the retrained NASNet. Precision, recall, the F1 score, and support as well as weighted precision, recall, and the F1 score are presented in Table B-3.
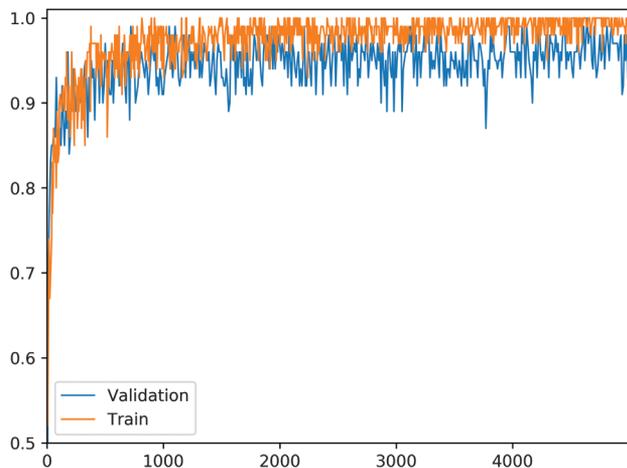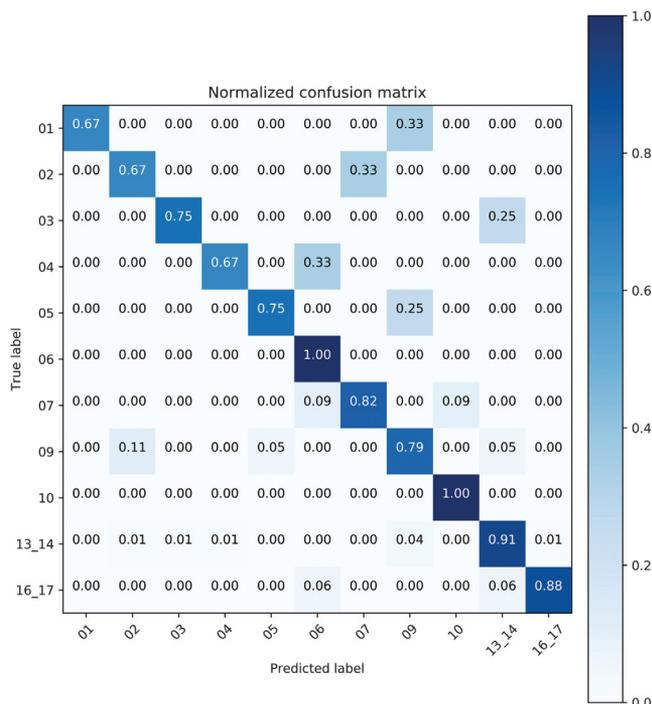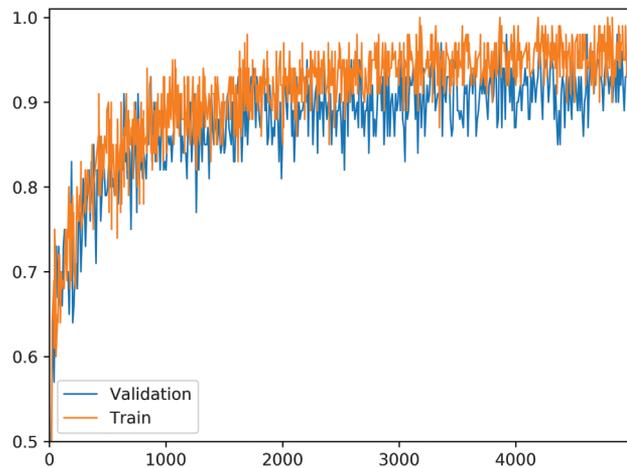

Figure B-3. Validation and training accuracy for the MobilNetV3 training. Note that after approximately 1000 iterations, the gains are marginal. Because the cost of training the classification layer is inexpensive, we can afford to let the model train for many steps.

Figure B-5. Validation and training accuracy for the NASNet training. Note that this architecture takes longer to increase its accuracy.
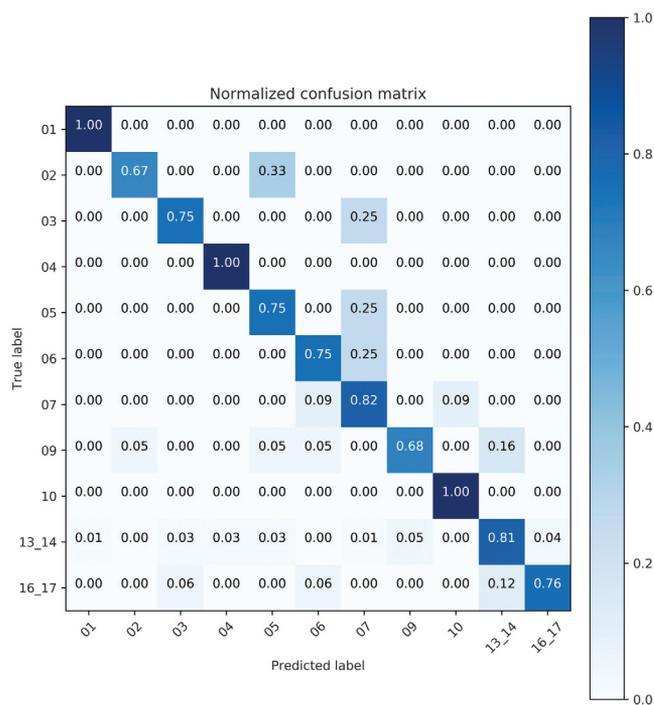

Figure B-6. Normalized confusion matrix of the retrained NASNet applied to the test set. Refer to Table 1 in the main text for the class lithofacies and the number of images for each class.

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 USENIX Symposium on Operating Systems Design and Implementation, 265–283.

Allen, D. B., and M. J. Pranter, 2016, Geologically constrained electrofacies classification of fluvial deposits:

Figure B-4. Normalized confusion matrix of the retrained MobileNetV2 applied to the test set. Refer to Table 1 in the main text for class lithofacies and the number of images for each class.

An example from the cretaceous Mesaverde Group, Uinta and Piceance Basins: AAPG Bulletin, **100**, 1775–1801, doi: 10.1306/05131614229.

Baldwin, J. L., R. M. Bateman, and C. L. Wheatley, 1990, Application of a neural network to the problem of mineral identification from well logs: Presented at the SPWLA 31.

Bestagini, P., V. Lipari, and S. Tubaro, 2017, A machine learning approach to facies classification using well logs: 87th Annual International Meeting, SEG, Expanded Abstracts, 2137–2142, doi: 10.1190/segam2017-17729805.1.

Blagoderov, V., I. J. Kitching, L. Livermore, T. J. Simonsen, and V. S. Smith, 2012, No specimen left behind: Industrial scale digitization of natural history collections: ZooKeys, **209**, 133–146, doi: 10.3897/zookeys.209.3178.

Buda, M., A. Maki, and M. A. Mazurowski, 2018, A systematic study of the class imbalance problem in convolutional neural networks: Neural Networks, **106**, 249–259, doi: 10.1016/J.NEUNET.2018.07.011.

Carranza-Rojas, J., H. Goeau, P. Bonnet, E. Mata-Montero, and A. Joly, 2017, Going deeper in the automated identification of Herbarium specimens: BMC Evolutionary Biology, **17**, 181, doi: 10.1186/s12862-017-1014-z.

Chollet, F., 2015, Keras, https://keras.io, accessed 15 May 2019.

Daily, M., S. Medasani, R. Behringer, and M. Trivedi, 2017, Self-driving cars: Computer, **50**, 18–23, doi: 10.1109/MC .2017.4451204.

de Matos, M. C., P. L. Osorio, and P. R. Johann, 2007, Unsupervised seismic facies analysis using wavelet transform and self-organizing maps: Geophysics, **72**, no. 1, P9–P21, doi: 10.1190/1.2392789.

de Matos, M. C., M. (Moe) Yenugu, S. M. Angelo, and K. J. Marfurt, 2011, Integrated seismic texture segmentation and cluster analysis applied to channel delineation and chert reservoir characterization: Geophysics, **76**, no. 5, P11–P21, doi: 10.1190/geo2010-0150.1.

Deng, J., W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, 2009, ImageNet: A large-scale hierarchical image database: IEEE Conference on Computer Vision and Pattern Recognition, 248–255, doi: 10.1109/CVPR.2009.5206848.

Dumoulin, V., and F. Visin, 2016, A guide to convolution arithmetic for deep learning: ArXiv e–prints.

Ellwood, E. R., B. A. Dunckel, P. Flemons, R. Guralnick, G. Nelson, G. Newman, S. Newman, D. Paul, G. Riccardi, N. Rios, K. C. Seltmann, and A. R. Mast, 2015, Accelerating the digitization of biodiversity research specimens through online public participation: BioScience, **65**, 383–396, doi: 10.1093/biosci/biv005.

Esteva, A., B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, 2017, Dermatologist-level classification of skin cancer with deep neural networks: Nature, **542**, 115–118, doi: 10.1038/nature21056.

Everson, R. M., and J. E. Fieldsend, 2006, Multi-class ROC analysis from a multi-objective optimisation perspective: Pattern Recognition Letters, **27**, 918–927, doi: 10 .1016/J.PATREC.2005.10.016.

Fawcett, T., 2006, An introduction to ROC analysis: Pattern Recognition Letters, **27**, 861–874, doi: 10.1016/J .PATREC.2005.10.010.

Ferri, C., J. Hernández-Orallo, and M. A. Salido, 2003, Volume under the ROC surface for multi-class problems BT, *in* N. Lavrač, D. Gamberger, H. Blockeel, and L. Todorovski, eds., Machine learning: ECML 2003: Springer, 108–120.

Gardner, J. W., T. C. Pearce, S. Friel, P. N. Bartlett, and N. Blair, 1994, A multisensor system for beer flavour monitoring using an array of conducting polymers and predictive classifiers: Sensors and Actuators B: Chemical, **18**, 240–243, doi: 10.1016/0925-4005(94)87089-6.

Gomez Villa, A., A. Salazar, and F. Vargas, 2017, Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks: Ecological Informatics, **41**, 24–32, doi: 10.1016/J.ECOINF.2017.07.004.

He, K., X. Zhang, S. Ren, and J. Sun, 2016, Identity mappings in deep residual networks, *in* B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., Computer vision — ECCV 2016: Springer International Publishing, 630–645.

Hunter, J. D., 2007, Matplotlib: A 2D graphics environment: Computing in Science and Engineering, **9**, 90–95, doi: 10 .1109/MCSE.2007.55.

Japkowicz, N., and S. Stephen, 2002, The class imbalance problem: A systematic study: Intelligent Data Analysis, **6**, 429–449.

Kortström, J., M. Uski, and T. Tiira, 2016, Automatic classification of seismic events within a regional seismograph network: Computers and Geosciences, **87**, 22–30, doi: 10.1016/J.CAGEO.2015.11.006.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, ImageNet classification with deep convolutional neural networks: Proceedings of the 25th International Conference on Neural Information Processing Systems, 1097–1105.

LeCun, Y., Y. Bengio, and G. Hinton, 2015, Deep learning: Nature, **521**, 436–444, doi: 10.1038/nature14539.

NOAA, 2016, Archive of core and site/hole data and photographs from the Deep Sea Drilling Project (DSDP), https://catalog.data.gov/dataset/archive-of-core-and-site-hole-data-and-photographs-from-the-deep-sea-drilling-project-dsdp, accessed 6 December 2018.

Norouzzadeh, M. S., A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, 2018, Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning: Proceedings of the National Academy of Sciences of the United States of America, **115**, E5716–E5725, doi: 10.1073/ pnas.1719367115.

Olah, C., A. Mordvintsev, and L. Schubert, 2017, Feature visualization: Distill, doi: 10.23915/distill.00007.

Olah, C., A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, 2018, The building blocks of interpretability: Distill, doi: 10.23915/distill.00010.

Oquab, M., L. Bottou, I. Laptev, and J. Sivic, 2014, Learning and transferring mid-level image representations using

convolutional neural networks: IEEE Conference on Computer Vision and Pattern Recognition, 1717–1724, doi: 10.1109/CVPR.2014.222.

Pan, S. J., and Q. Yang, 2010, A survey on transfer learning: IEEE Transactions on Knowledge and Data Engineering, **22**, 1345–1359, doi: 10.1109/TKDE.2009.191.

Perol, T., M. Gharbi, and M. Denolle, 2018, Convolutional neural network for earthquake detection and location: Science Advances, **4**, e1700578, doi: 10.1126/sciadv .1700578.

Qi, J., T. Lin, T. Zhao, F. Li, and K. Marfurt, 2016, Semisupervised multiattribute seismic facies analysis: Interpretation, **4**, no. 1, SB91–SB106, doi: 10.1190/INT-2015-0098.1.

Qian, F., M. Yin, X.-Y. Liu, Y.-J. Wang, C. Lu, and G.-M. Hu, 2018, Unsupervised seismic facies analysis via deep convolutional autoencoders: Geophysics, **83**, no. 3, A39–A43, doi: 10.1190/geo2017-0524.1.

Roy, A., A. S. Romero-Peláez, T. J. Kwiatkowski, and K. J. Marfurt, 2014, Generative topographic mapping for seismic facies estimation of a carbonate wash, Veracruz Basin, southern Mexico: Interpretation, **2**, no. 1, SA31–SA47, doi: 10.1190/INT-2013-0077.1.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, 2015, ImageNet large scale visual recognition challenge: International Journal of Computer Vision, **115**, 211–252, doi: 10.1007/s11263-015-0816-y.

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, 2018, MobileNetV2: Inverted residuals and linear bottlenecks: ArXiv e–prints.

Sener, O., and S. Savarese, 2018, Active learning for convolutional neural networks: A core-set approach: International Conference on Learning Representations.

Settles, B., 2012, Active learning: Synthesis Lectures on Artificial Intelligence and Machine Learning, **6**, 1–114, doi: 10.2200/S00429ED1V01Y201207AIM018.

Simonyan, K., and A. Zisserman, 2014, Very deep convolutional networks for large-scale image recognition, https://arxiv.org/abs/1409.1556, accessed 15 May 2019.

Sinha, S., Y. Wen, R. A. Pires de Lima, and K. Marfurt, 2018, Statistical controls on induced seismicity: Unconventional Resources Technology Conference, doi: 10 .15530/urtec-2018-2897507-MS.

Suriamin, F., and M. J. Pranter, 2018, Stratigraphic and lithofacies control on pore characteristics of Mississippian limestone and chert reservoirs of north-central Oklahoma: Interpretation, **6**, no. 4, T1001–T1022, doi: 10 .1190/int-2017-0204.1.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, 2014, Going deeper with convolutions: CoRR, abs/ 1409.4.

Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, 2015, Rethinking the inception architecture for computer vision: CoRR, abs/1512.0.

Tajbakhsh, N., J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, 2016, Convolu-

tional neural networks for medical image analysis: Full training or fine tuning?: IEEE Transactions on Medical Imaging, **35**, 1299–1312, doi: 10.1109/TMI.2016.2535302.

TensorFlow Hub, 2018, https://tfhub.dev/, accessed 26 November 2018.

USGS Core Research Center, 2018, Links to core repositories, https://www.usgs.gov/core-science-systems/nggdp/ core-research-center/links-core-repositories, accessed 6 December 2018.

Wu, Y., Y. Lin, Z. Zhou, D. C. Bolton, J. Liu, and P. Johnson, 2018, Deepdetect: A cascaded region-based densely connected network for seismic event detection: IEEE Transactions on Geoscience and Remote Sensing, **57**, 1–14, doi: 10.1109/TGRS.2018.2852302.

Yosinski, J., J. Clune, Y. Bengio, and H. Lipson, 2014, How transferable are features in deep neural networks?: Advances in Neural Information Processing Systems, **27**, 3320–3328.

Zhang, G., Z. Wang, H. Li, Y. Sun, Q. Zhang, and W. Chen, 2018, Permeability prediction of isolated channel sands using machine learning: Journal of Applied Geophysics, **159**, 605–615, doi: 10.1016/J.JAPPGEO.2018.09.011.

Zhang, Y., H. A. Salisch, and J. G. McPherson, 1999, Application of neural networks to identify lithofacies from well logs: Exploration Geophysics, **30**, 45, doi: 10 .1071/EG999045.

Zhao, T., F. Li, and K. J. Marfurt, 2017, Constraining self-organizing map facies analysis with stratigraphy: An approach to increase the credibility in automatic seismic facies classification: Interpretation, **5**, no. 2, T163–T171, doi: 10.1190/INT-2016-0132.1.

Zhao, T., J. Zhang, F. Li, and K. J. Marfurt, 2016, Characterizing a turbidite system in Canterbury Basin, New Zealand, using seismic attributes and distance-preserving self-organizing maps: Interpretation, **4**, no. 1, SB79–SB89, doi: 10.1190/INT-2015-0094.1.

Zoph, B., and Q. V. Le, 2016, Neural architecture search with reinforcement learning: CoRR, abs/1611.0.

Zoph, B., V. Vasudevan, J. Shlens, and Q. V. Le, 2017, Learning transferable architectures for scalable image recognition: CoRR, abs/1707.0.

Rafael Pires de Lima received a bachelor's degree in geophysics from the Universidade de São Paulo, Brazil, and an M.S. in geophysics from the University of Oklahoma. He is currently on leave from the Geological Survey of Brazil to pursue his doctorate in geophysics and a second M.S. in data science and analytics at the University of Oklahoma. He has industrial experience in seismic processing and nonseismic methods. His research interests include image processing, seismic attributes, and machine-learning applications for geoscience research.

Fnu Suriamin received a B.E in geological engineering from Padjadjaran University and an M.Sc. in geology from the Colorado School of Mines. He worked in Core Lab and Halliburton Sperry Sun Drilling Services in Jakarta. He previously worked as a geoscientist/petrophysicist with ExxonMobil in Jakarta and Petrofac Malaysia Ltd. in Kuala Lumpur. Currently, he is pursuing his Ph.D. at the University of Oklahoma and is a petroleum geologist and petrophysicist at the Oklahoma Geological Survey. His research interests include petrophysics, reservoir characterization and modeling, mathematical geology, and petroleum geology.

Kurt J. Marfurt began his geophysical career teaching geophysics and contributing to an industry-supported consortium on migration, inversion, and scattering (project MIDAS) at Columbia University's Henry Krumb School of Mines in New York City. In 1981, he joined Amoco's Tulsa Research Center and spent the next 18 years leading research efforts in modeling, migration, signal analysis, basin analysis, seismic attribute analysis, reflection tomography, seismic inversion, and multicomponent data analysis. In 1999, he joined the University of Houston as a professor in the geoscience department and as a director of Allied Geophysics Laboratories. He is currently a member of the Geophysical Societies of Tulsa and Houston, SEG, EAGE, AAPG, AGU, and SIAM, and he serves as Deputy Editor-in-Chief of *Interpretation*. His current research activity includes prestack imaging, velocity analysis and inversion of converted waves, computer-assisted pattern recognition of geologic features on 3D seismic data, and interpreter-driven seismic processing. His research interests include seismic signal analysis, 3D seismic attributes, seismic velocity analysis, subsurface imaging, and multicomponent data analysis.

Matthew J. Pranter received a B.S. in geology from Oklahoma State University, a B.S. in geological engineering from the Colorado School of Mines, an M.S. in geology from Baylor University, and a Ph.D. in geology from the Colorado School of Mines. He holds the Lew and Myra Ward endowed chair in reservoir characterization and is a professor of geology and geophysics at the University of Oklahoma. He previously worked as a geology professor at the University of Colorado at Boulder and a senior research/reservoir geologist with ExxonMobil Upstream Research Company and Conoco. His research interests include reservoir characterization and modeling, sedimentary geology, and petroleum geology.