ABSTRACT

The recent interest in using deep learning for seismic interpretation tasks, such as facies classification, has been facing a significant obstacle, namely the absence of large publicly available annotated datasets for training and testing models. As a result, researchers have often resorted to annotating their own training and testing data. However, different researchers may annotate different classes, or use different train and test splits. In addition, it is common for papers that apply deep learning for facies classification to not contain quantitative results, and rather rely solely on visual inspection of the results. All of these practices have lead to subjective results and have greatly hindered the ability to compare different machine learning models against each other and understand the advantages and disadvantages of each approach.

To address these issues, we open-source an accurate 3D geological model of the Netherlands F3 Block. This geological model is based on both well log data and 3D seismic data and is grounded on the careful study of the geology of the region. Furthermore, we propose two baseline models for facies classification based on deconvolution networks and make their codes publicly available. Finally, we propose a scheme for evaluating different models on this dataset, and we share the results of our baseline models. In addition to making the dataset and the code publicly available, this work can help advance research in this area and create an objective benchmark for comparing the results of different machine learning approaches for facies classification for researchers to use in the future.

INTRODUCTION

In recent years, there has been great interest in using deep learning for seismic interpretation tasks such as facies classification (Shi et al., 2018; Dramsch and Lthje, 2018; Waldeland and Solberg, 2017; Araya-Polo et al., 2017; Huang et al., 2017; Zhao, 2018; Di et al., 2018; Rutherford Ildstad and Bormann, 2017). Typically, deep learning models — such as convolutional neural networks (CNNs) — have millions of free parameters, and therefore require a large amount of annotated training data. Unfortunately, and unlike other areas of research such as computer vision, there is a lack of large publicly-available annotated datasets for seismic interpretation that can be used to train and benchmark machine learning models. To address this problem, some researchers resort to annotating their own training and testing dataset. For example, in the Netherlands F3 block, Zhao (2018) annotated 40 inlines, Di et al. (2018) annotated 12 inlines, while Rutherford Ildstad and Bormann (2017) only annotated a single inline for their model. The limited number of annotated sections in understandable given that the annotation process is time-consuming, requires subject matter expertise, and can be quite subjective. Nevertheless, such limited annotation undermines the mass potential machine learning could have when deployed in such a field.

Alternatively, there has been some research in attempting to avoid annotating large amounts of data by using weakly-supervised learning approaches. Alaudah and AlRegib (2016) trained a facies classification model using seismic images with image-level labels only. Later, Alaudah et al. (2019) proposed a method for generating large amounts of training data using similarity-based retrieval and a weakly-supervised label mapping algorithm. As few as one or two exemplar images per class were enough to automatically generate a large amount of training data. These automatically-generated training data was then used to train a weakly-supervised deconvolution network (Alaudah et al., 2018) for facies classification. Other researchers avoid supervision all-together by using...
traditional unsupervised machine learning techniques such as principal component analysis or self-organizing maps. There is a very rich literature on traditional supervised and unsupervised methods for facies classification. [Zhao et al., 2015] provides an excellent review of some of the most commonly used techniques. More recently, unsupervised techniques based on deep learning models such as deep convolutional autoencoders have been explored (Qian et al., 2018; Shafiq et al., 2018).

Whether researchers annotate their own training data, or use other techniques, there still remains a lack of large publicly-available annotated datasets for seismic interpretation that can be used for training different models and comparing the performance of different approaches. Furthermore, it is common for papers that apply deep learning for facies classification, or other seismic interpretation tasks, to not contain quantitative results, but rather rely solely on subjective visual inspection of the results. All of this leads to highly subjective results and greatly hinders the ability of researchers to compare different approaches against each other and understand the advantages and disadvantages of each approach.

To address these issues, and to help make machine learning research in seismic interpretation more reproducible, we open-source a fully-annotated 3D geological model of the Netherlands F3 Block [dGB Earth Sciences, 1987]. This model is grounded in the geology of the region and based on the study of both 3D seismic data and well log data, and not only the visual appearance of the seismic data. The data also includes fault planes that we have extracted from the F3 block. Furthermore, we also present two baseline models for facies classification based on a deconvolution network architecture. The first baseline is a patch-based model that is trained using a large number of small patches extracted from all the inlines and crosslines in the training set. The second baseline is a section-based model that was trained directly on entire crosslines in the training set. The second baseline is a patch-based model that is trained using a large number of small patches extracted from all the inlines and crosslines in the training set. In addition, we have open-sourced all the codes that was used to train and test our baseline models using the PyTorch deep learning library.[1] Finally, we propose a common procedure for evaluating different models on this dataset, and we share the results for our baseline models. The next section provides a quick overview of the geology of the Netherlands F3 block, and introduces our geological model.

A 3D GEOLOGICAL MODEL OF THE NETHERLANDS F3 BLOCK

The North Sea is rich in hydrocarbon deposits, which is why this area is very well studied in the literature [Door-nembal, 2014]. The North Sea continental shelf, located off the shores of the Netherlands is divided into geographical zones described by different letters of the alphabet, within these zones are smaller areas marked with numbers. One of these areas is a rectangle of dimensions 16 km x 24 km known as the F3 block, see Figure[1]. In 1987, the F3 block 3D seismic survey was conducted to identify the geological structures of this area and to search for hydrocarbon reservoirs. In addition, many boreholes were drilled within the F3 block throughout the years. The F3 block became one of the most widely known and studied seismic surveys after dGB Earth Sciences made the data obtained from the survey publicly available.

The aim of this section is to briefly describe the geology of the survey area and introduce the 3D geological model that we have developed and how it was obtained.

The geology of the F3 block

Within the shelf of the North Sea ten lithostratigraphic units have been identified in the literature (Van Adrichem Bo-gaert and Kouwe, 1993; Mijnlieff, 2002; Scheck-Wenderoth and Lamarche, 2005; Duin et al., 2006). These units and their main lithostratigraphic features are listed below from newest to oldest:

1. Upper North Sea group: claystones and sandstones from Miocene to Quaternary.
2. Lower and Middle North Sea groups: sands, sandstones, and claystones from Paleocene to Miocene.
3. Chalk group: carbonates of Upper Cretaceous and Paleocene.
4. Rijnland group: clay formations with sandstones of Upper Cretaceous.
5. Schieland, Scruff and Niedersachsen groups: claystones of Upper Jurassic and Lower Cretaceous.
6. Altena group: claystones and carbonates of Lower and Middle Jurassic.
7. Lower and Upper Germanic Trias groups: sandstones and claystones of Triassic.
8. Zechstein group: evaporates and carbonates of Zechstein.
9. Upper and Lower Rotliegend groups: siliceous rocks and basalts of the Lower Zechstein.
10. Limburg group: Upper carboniferous siliceous rock, which are the bedrock for hydrocarbons.

The F3 block is located on the border of two tectonic structures: Step Graben and Dutch Central Graben (see Figure[2]), which are characterized by different appearance and varying thickness of lithostratigraphic units. This diversity is a result of tectonic activity (Ziegler, 1988; 1990), which was started in Variscan orogeny (Schroot and De Haan, 2003). The area within the Step Graben is strongly disturbed by salt diapirs, which were active several times, from Zechstein to Paleogene period [Remmelts]
On the other hand, only in Dutch Central Graben, as a result of subsiding Jurassic rocks of the Altena group and Scruff, Schieland and Niedersachsen groups are observed (Duin et al., 2006).

The modelling process

To prepare our 3D geological model of the F3 block, we relied on both well logs and 3D seismic data. The next two subsections describe the process of creating the model using both the well logs and the 3D seismic data.

3D model building using well logs data

The well log data were obtained from a website managed by the Geological Survey of the Netherlands (www.nlog.nl). The data (including information related to coordinates, true vertical depth, measured depth along the curvature, inclinations, and individual horizons) were collected for 26 boreholes located within the F3 block or its vicinity. The exact locations of these wells are visualized in Figure 3.

Originally, the 26 wells contained 40 different horizons, so it was necessary to distill them to the ten lithostratigraphic units adopted in literature and presented in the previous subsection. The next step was correlating wells with each other. After that, it was possible to create a preliminary 3D model based on the well log data by using Petrel’s make/edit surface tool. This process facilitated the preliminary visualization of the range of individual horizons, which was very helpful in the further interpretation of the 3D seismic data.

3D model building using seismic data

The F3 block data was migrated in time, not depth, so it was necessary to do time-depth conversion since the structural model must be prepared in the depth domain. OpendTect 5.0 was used to perform the time-depth conversion using a velocity model that was provided with the F3 block data. After the process of time-depth conversion, the 3D seismic data was imported to Petrel and viewed with the wells and interpreted horizons in order to compare the coverage of the seismic data and the wells in terms of surface and depth.

The next step in creating the model was faults-surface interpretation. Using Petrel’s polygon editing tool, we interpreted the main fault surfaces and the fault networks.
were created by using the fault framework modeling tool in Petrel. Horizons were interpreted in a similar fashion, but by using the seeded 3D autotracking tool, which interpolated data automatically and took into account the faults networks modeled previously.

Based on the interpreted horizons and faults preliminary modeling was conducted. This was done using the horizon modeling tool with volume-based modeling, which is an advanced method of isochronous geological space modeling. The preliminary model included several imperfections in the interpretations within horizons and faults, so it was necessary to re-model several faults and conduct small corrections in the interpreted horizons.

After this, it was possible to create the final three-dimensional model which highlights the regions between individual horizons. Here, Petrel’s structural modeling module in the horizon modeling tool was used in addition to the create zone model function. The final 3D geological model is shown in Figure 4.
The 3D geological model

Within our 3D geological model of the F3 Block on the shelf of the North Sea in the Netherlands, seven geological units have been identified (see Figure 4). There are (from newest to oldest): Upper North Sea group, Middle North Sea group, Lower North Sea group, Chalk group, Rijnland group, Scuff group, and Zechstein group.

These units can be divided into three structural levels: **Cenozoic** (Lower, Middle, and Upper North Sea groups), **Mesozoic** (Scuff, Rijnland, and Chalk groups), and **Permian** (Zechstein group).

As is evident in Figure 4, the F3 Block is characterized by highly variable geological structures, both in the horizontal and vertical range, which is manifested by the differential thicknesses of individual units and by the expanded faults network related to salt tectonics. The area of the F3 Block can be divided into 2 regions: Eastern and Western. The Eastern region is disturbed by the occurrence of Zechstein diapirs and irregular faults network. The Western region is characterized by regular fault networks and a more uniform thickness of lithostratigraphic units.

The **Upper North Sea group** is the youngest and the most lay lithostratigraphic unit within our model. The top of the Upper North Sea group is the bottom of the North Sea at the same time, which is about -40 meters above sea level (m a.s.l). Differences in the depth of the ocean floor are small and they are maximally 6 meters within the whole F3 Block. It can be noted that the depth of this top decreases from SW to NE. The thickness of the Upper North Sea group varies from about 1000 m (in places deformed by Permian diapirs) to about 1320 m in the northern part of the research area (see Figure 4).

Below the Upper North Sea group lays the **Middle North Sea group**. The depth of the top of this unit ranges from -1000 m a.s.l. within the diapir in NE part of F3 Block to about -1360 m a.s.l. in the northern part of this area, between diapirs. The thickness of the Middle North Sea group is from 20 to 150 m. As in the case of the Upper North Sea group, the relationship between the occurrence of Zechstein salts and the depth of the top and the thickness of this unit is visible. Clearly visible are differences of thickness between both sites of faults also.

The next unit is the **Lower North Sea group**. This unit has an analogous character to the Middle North Sea...
group. The top is at a depth from -1100 m a.s.l., while the thickness is from about 180 to 750 m.

The top of the Chalk group is at depth from -1300 m a.s.l. (above the diapirs in the NE part of the survey) to -2100 m a.s.l. (in the Eastern part of the survey, which is undisturbed by diapirs). The minimum thickness of this unit is 25 m, while above the salt diapirs in NE part of F3 Block, this substantially increases to 525 m.

The top surface of the Rijnland group is submerged in the NNE direction, while it is the shallowest in the NE part of F3 Block and above the Zechstein diapirs (about -1500 m a.s.l.). The thickness of the Rijnland group is very diverse. The maximum value is about 200 m (above diapirs), while in the other parts of F3 block it is less than 20 m or does not occur at all.

The Scruff group, similar to Rijnland group, is thinned out in NNE direction, more or less in the middle of F3 Block, where the top of this layer has a depth of -2180 m a.s.l. This layer is the shallowest (-1500 m a.s.l.) in the SW part of the F3 block and above the Zechstein diapirs in the Southern part of the survey. The thickness of the Scruff group within our model boundaries ranges from 100 m to almost 700 m.

The Zechstein group occurs only in the eastern part of the survey, as irregularly-shaped salt diapirs. The shallowest part of the Zechstein group is at a depth of -1500 m a.s.l. while the maximum thickness of the Zechstein group within the research area is about 700 m. However, like in the case of the Scruff group, the depth is much bigger. According to the literature, it can reach several kilometers [Duin et al. 2006].

Figure 5: An overhead view of 3D fault planes from three different generations of faults that we have identified in the F3 block.

In addition to the identified lithostratigraphic units mentioned above, we have also identified three generations of faults. The first generation are reverse, oblique-slip, sinistrals faults with a SSW-NNE orientation. This direction is connected with the course of the tectonic axis of the Dutch Central Graben, which (similar to the whole Graben) has an SSW-NNE orientation. The second generation of faults are normal, oblique-slip, dextral faults with a W-E orientation. Finally, the third generation are faults disturbed by Permian halokinesis; they are genetically linked with faults from the first and second generation. Figure 5 shows an overhead view of the three generations of faults that we have identified.

DECONVOLUTION NETWORK BASELINE

In addition to the geological model we’ve introduced in the last section, we propose two baseline models for facies classification based on a deconvolution network architecture. In this section, we introduce deconvolution networks and describe our two baseline models.

Deconvolution networks

Convolutional neural networks (CNNs) have seen great success in a wide number of visual applications, from image classification to semantic labeling. In the previous few years, it was not well established that end-to-end CNN architectures can perform very well in semantic labeling tasks (such as facies classification). A major hurdle to the success of end-to-end CNN architectures in these tasks was what seemed like a trade-off between classification and localization accuracy. Deeper networks that have many convolution and pooling layers have proven to be the most successful models in image classification tasks. However, their large receptive fields and increased spatial invariance (due to pooling and convolutional layers) make it difficult to infer the locations of various objects within the im-
age. In other words, the deeper we go into a network, the more it seemed we lose the location information of various objects within the image. Some researchers have attempted to overcome this hurdle by using various pre- or post-processing techniques. However, the introduction of fully convolutional network architectures, such as FCN (Long et al., 2015) and DeconvNet (Noh et al., 2015), have shown that it is possible to achieve good semantic labeling results using a convolutional network only, with no pre- or post-processing steps required. FCN accomplish this by replacing the fully-connected layers of the CNN with 1D convolutional layers that produce coarse feature maps. These coarse feature maps are then upsampled, and concatenated with the scores from intermediate feature maps in the network to generate the output. These upsampling steps, however, result in a blurred output that loses some of the resolution of the original image.

Deconvolution networks, on the other hand, overcome this problem by using a symmetric encoder-decoder style architecture composed of stacks of convolution and pooling layers in the encoder, and stacks of deconvolution and unpooling layers in the decoder that mirror the encoders architecture. The role of the encoder can be seen as doing object detection and classification, while the decoder is used for accurate localization of these objects within the image. This architecture can achieve finer and more accurate results than those of the FCN, and therefore is adopted in our work.

A few recent papers have illustrated the successful application of deconvolution networks for seismic interpretation applications (Alaudah et al., 2018; Di et al., 2018).

Figure 7 illustrates the architecture of the deconvolution network used for both of our baseline models. Every convolution or deconvolution layer (in white) is followed by a rectified linear unit (ReLU) non-linearity. The layers in red perform $2 \times 2$ max pooling to select the maximum filter response within small windows. The indices of the maximum responses for every pooling layer are then shared with their respective unpooling layers (in green) to undo this pooling operation and get a higher resolution image.

**Baseline Models**

In this work, we use two baseline models; a patch-based and a section-based model. These two models use the exact same architecture, optimizer, and hyperparameters but differ in the way they are trained and the way they are used to label the seismic volume.

**Patch-based model:**

The patch-based model is trained on small patches extracted from the inlines and crosslines of the training data. For very large seismic volumes, this approach can be more feasible than using entire sections for training. At training time, the patches of seismic data and their associated labels are sampled randomly from the inlines and crosslines of the training set. During test time, the model samples overlapping patches in the inline and crossline direction and average the results to generate a 2D labeled version of the test inline or crossline. This is done for all inlines and crosslines in the test sets.

**Section-based model:**

The section-based model is trained on entire inline and crossline sections. The advantage of this approach is twofold. First, since the network is fed an entire section, it can easily learn the relationships between different lithostratigraphic units and can take the depth information into account when labeling the section. The second advantage is more practical. Training and testing entire sections at once means the network can be trained or tested very quickly since there are only a relatively small number of seismic inlines and crosslines. One advantage of using a fully convolutional architecture (such as the one we are using) is that the size of the network input does not have to be fixed. The size of the output of the network changes as the size of its input change. Therefore, the different size of the inline and crossline sections does not pose any

\[1\] This is assuming the GPU memory is large enough to handle the size of the seismic sections. On our NVIDIA Titan X GPU, we trained the baseline section-based network – eight sections at a time – in about 70 minutes.
problem to the training of this network.

**Other variations:**

In addition to the baseline patch- and section-based models, we have trained other variations of these models to test how they can be improved. We have tested the following variations:

- **Baseline + data-augmentation:** data augmentation applies different label-preserving transformations to the training data such as rotation, random horizontal flipping, and the addition of Gaussian noise. This can help increase the training sample size, and help the network generalize better to the test data.

- **Baseline + skip connections:** In a deep neural network, the output of a layer is typically passed on as the input to the next layer in the network. Skip connections allow the output of a layer to be also passed as an input to a layer farther up the network, skipping intermediate layers in the process. These connections are implemented by directly adding the feature maps of various layers in the encoder part of the deconvolution network to the feature maps of the corresponding layers in the decoder. Skip connections help networks overcome the vanishing gradient problem [Hochreiter et al., 2001] by providing “shortcuts” for the computed gradients to propagate to the lower layers of the network.

**EXPERIMENTAL SETUP**

In this section, we will introduce the main elements of the experimental setup, including how the final geological model was produced, how the model is split into training and testing sets, and what metrics are used to objectively evaluate the performance.

**The geological model**

The final geological model that we use to train and test our models is not the entire volume shown in Figure 4. The time-depth conversion process of the seismic data resulted in some artifacts. These artifacts were concentrated along the sigmoidal structure in the Upper North Sea group. Due to these artifacts, and missing data on the sides of the survey, we only use the data between inlines 100 and 701, crosslines 300 and 1201, and depth between 1005 and 1877 meters. Furthermore, we combine the Rijnland and Chalk groups in our final model to a single class due to various issues with processing the Rijnland/Chalk boundary when generating the final model. Table 1 shows the percentage of different classes in our training set.

In addition to the final model labels and seismic data, we also release the original horizons for all the lithostratigraphic units, in addition to the extracted fault planes from all three generations.

**The train/test split**

Careful selection of the training and testing sets is crucial in any machine learning application. This is especially true in seismic data, where neighboring sections are highly correlated. Selecting the training and testing sections randomly will lead to artificially good test results, that are not representative of the actual generalization performance of the tested models. Therefore, it is important to minimize the correlation between the training and testing sets as much as possible. It is also important to ensure that both the training and testing sets have adequate representation of all the classes in the dataset.

Therefore, we decide to split the data as in shown in Figure 8. Namely, the data is split into the following three sets:

1. **Training set:** This includes all the data in the range of inlines [300,701] and crosslines [300,1000].
2. **Testing set 1:** This set includes all the data in the range of inlines [100-299] and crosslines [300-1000].
3. **Testing set 2:** This sets includes all the data in the ranges of inlines [100:701] and crosslines [1001-1200]. This set includes a large Zechstein diapir in the NE of the survey that is never seen in the training set.

For a fair comparison with others, it is important to note that the test sets should never be used more than once. Testing a model on the test set, then retraining that model or another model with different parameters means that the test set has been used for validation, which defeats its purpose.

**Evaluation metrics**

To objectively evaluate the performance of different models on our two test sets, we use the following metrics: pixel accuracy (PA), class accuracy (CA) for each individual class, mean class accuracy (MCA) for all classes, and frequency-weighted intersection over union (FW-IoU). These metrics are detailed in the appendix.

**RESULTS**

After the final geological model was created, we train each of the models described earlier on the training set until its training loss converges. We note that on our NVIDIA Titan X GPU, the baseline patch-based model, and the augmented version, converged after 16 hours of training. The patch-based model with skip connections required less than 5 hours to converge. While the section-based models required significantly less time, with all of them converging in less than 90 minutes. We test these models by using them to label all inlines and crosslines in both test sets, and computing the performance metrics on the final result. Table 2 summarizes the objective results for all the models that we have tested, while Figure 9 shows inline
Figure 8: A 3D view of the F3 block from above with the Zechstein Group shown in red, while the Chalk Group is shown in a semi-transparent beige color. Inline 300 and crossline 1000 divide the survey into four regions. The NW region of the survey is used for training, while the SW region constitutes the first test set. The remaining region East of crossline 1000 constitutes the second test set.

Table 1: The percentage of pixels from different classes in the training set.

| Class                | Percentage |
|----------------------|------------|
| Zechstein            | 1.48%      |
| Scruff               | 3.17%      |
| Rijnland/Chalk       | 6.53%      |
| Lower N. S.          | 48.44%     |
| Middle N. S.         | 11.89%     |
| Upper N. S.          | 28.49%     |

151 of test set 1 labeled using the six different models we have tested. In the remainder of this section, we will discuss these results, and suggest various methods to improve upon them.

Patch-based vs section-based models:

Since the patch-based models are trained on patches from different depths in the data, they can easily confuse various classes that typically exist at different depths. For example, the patch-based models in Figure 9 often confuse the Scruff group in the bottom left of the image with Lower, Middle, or Upper North Sea groups. While the section-based models do not suffer from these problems as often. In addition, it seems the patch-based models classify the North Sea groups fairly well, mainly due their distinct texture and their abundance in the training set. However, they perform very poorly on the other classes. On the other hand, the section-based models perform much better, while still facing problems with classifying smaller classes such as Zechstein and Scruff groups.

Imbalanced classes:

As Table 1 shows, our dataset is highly imbalanced. The Zechstein and Scruff groups are far smaller than the Lower or Upper North Sea groups. This means while training, the network is trained on far more examples of Lower or Upper North Sea groups than of Zechstein or Scruff groups for example. This leads to the networks being biased towards classifying pixels as Lower or Upper North Sea groups, and therefore artificially achieve high CA scores for those classes. However, this is at the expense of the very poor performance for the smaller classes.

We do not make any changes to the baseline models to overcome this issue. However, using various techniques to overcome this class imbalance can significantly improve the results, especially for the smaller classes, such as Zech-
Table 2: Results of our two baseline models, with other variations, when tested on both test splits of our dataset. All metrics are in the range $[0, 1]$, with larger values being better. The best performing model for every metric is highlighted in bold.

| Model                        | Metric   | PA  | Class Accuracy | MCA | FW-IoU |
|------------------------------|----------|-----|----------------|-----|--------|
|                              |          |     | Zechstein      | Scruff | Rijnland/Chalk | Lower N. S. | Middle N. S. | Upper N. S. |     |        |
| Patch-based model            |          | 0.788 | 0.118 | 0.042 | 0.423 | **0.988** | 0.807 | 0.807 | 0.531 | 0.638 |
| Patch-based + aug.           |          | 0.794 | 0.016 | 0.007 | 0.311 | 0.979   | 0.736 | 0.941 | 0.498 | 0.641 |
| Patch-based + skip conn.     |          | 0.841 | 0.436 | 0.226 | 0.663 | 0.966   | 0.897 | 0.884 | 0.678 | 0.729 |
| Section-based model          |          | 0.883 | 0.527 | 0.387 | 0.766 | 0.974   | 0.823 | **0.981** | 0.743 | 0.798 |
| Section-based + aug.         |          | 0.874 | 0.479 | 0.542 | 0.657 | 0.945   | 0.862 | 0.959 | 0.741 | 0.798 |
| Section-based + skip conn.   |          | **0.930** | **0.598** | **0.762** | **0.788** | 0.975 | **0.920** | 0.975 | **0.836** | **0.875** |

Figure 9: The results of the different models on inline 151 from test set 1.
stein and Scruff groups.

Data augmentation:

Data augmentation is a technique to artificially increase the size of the training set. This is quite useful when training a large network with a limited amount of training data. However, in our case, data augmentation exacerbates the class imbalance problem and therefore, leading to worse results in smaller classes such as Zechstein, while only slightly improving the results of larger classes. In some cases, larger classes such as the Lower North Sea group are not improved with data augmentation. This highlights the importance of having a balanced training dataset.

Skip connections:

For both patch- and section-based models, using skip connections can greatly improve the results, and speed up the training. This is especially noticeable in smaller classes. For example, adding skip connections improved the class accuracy for the Scruff group in the section-based model. Furthermore, the skip connection models outperformed the baseline by more than 8% on the FW-IoU metric. Overall, adding skip connections seems to greatly help improve the results, especially for smaller classes. It also can potentially speed up to the training process. In the case of the patch-based model, the skip connection model converged four times faster than the baseline.

CONCLUSIONS

In conclusion, we have introduced and made publicly available a new annotated dataset for facies classification. This dataset includes six different lithostratigraphic classes based on the underlying geology of the Netherlands F3 block. The dataset also includes fault planes from three different generations that we have identified in the F3 block.

In addition, we present two baseline deep learning models for facies classification, a patch- and a section-based model, both based on a deconvolution network architecture. We train these models using our dataset, and we analyze their performance. Furthermore, we make the code for training and testing these models publicly available for others to use.

It is our hope that this dataset, and the code that we have released, will help facilitate more research in this area, and help create an objective benchmark for comparing the results of different machine learning approaches for facies classification.

ACKNOWLEDGEMENTS

We would like to acknowledge the support of the Center for Energy and Geo Processing (CeGP) at the Georgia Institute of Technology and King Fahd University of Petroleum and Minerals (KFUPM). We would also like to thank dGB Earth Sciences for making the Netherlands F3 data publicly available, and Schlumberger for providing educational licenses for the Petrel software.

APPENDIX A

EVALUATION METRICS

To objectively evaluate the performance of our models on this dataset, we use a set of evaluation metrics that are commonly used in the computer vision literature. If we denote the set of pixels that belong to class \( i \) as \( G_i \), and the set of pixels classified as class \( i \) as \( F_i \). Then, the set of correctly classified pixels is simply \( G_i \cap F_i \). We use \(| \cdot |\) to denote the number of elements in a set. Now, we can define the following metrics:

- **Pixel Accuracy (PA)** is the percentage of pixels over all classes that are correctly classified,

\[
PA = \frac{\sum_i |F_i \cap G_i|}{\sum_i |G_i|}, \quad (A-1)
\]

- **Class Accuracy for class \( i \) (CA_{i})** is the percentage of pixels that are correctly classified in a class \( i \).

\[
CA_i = \frac{|F_i \cap G_i|}{|G_i|}, \quad (A-2)
\]

We will also define the **Mean Class Accuracy (MCA)** as the average of CA over all classes,

\[
MCA = \frac{1}{n_c} \sum_i CA_i = \frac{1}{n_c} \sum_i \frac{|F_i \cap G_i|}{|G_i|}, \quad (A-3)
\]

where \( n_c \) is the number of classes.

- **Intersection over Union (IoU_{i})** is defined as the number of elements of the intersection of \( G_i \) and \( F_i \) over the number of elements of their union set,

\[
\text{IoU}_{i} = \frac{|F_i \cap G_i|}{|F_i \cup G_i|}, \quad (A-4)
\]

This metric measures the overlap between the two sets and it should be 1 if and only if all pixels were correctly classified. Further, when we average IoU over all classes, we arrive at the **Mean Intersection over Union (Mean IoU)**,

\[
\text{Mean IoU} = \frac{1}{n_c} \sum_i \text{IoU}_{i} = \frac{1}{n_c} \sum_i \frac{|F_i \cap G_i|}{|F_i \cup G_i|}. \quad (A-5)
\]

To prevent this metric from being overly sensitive to small classes, it is common to weigh each class by its size. The resulting metric is known as **Frequency-Weighted Intersection over Union (FW-IoU)**,

\[
\text{FW-IoU} = \frac{1}{\sum_i |G_i|} \cdot \sum_i |G_i| \cdot \frac{|F_i \cup G_i|}{|F_i \cap G_i|}. \quad (A-6)
\]
REFERENCES

Alaudah, Y., M. Alfarraj, and G. AlRegib, 2019, Structure label prediction using similarity-based retrieval and weakly supervised label mapping: GEOPHYSICS, 84, V67–V79.

Alaudah, Y., and G. AlRegib, 2016, Weakly-supervised labeling of seismic volumes using reference exemplars: 2016 IEEE International Conference on Image Processing (ICIP), 4373–4377.

Alaudah, Y., S. Gao, and G. AlRegib, 2018, in Learning to label seismic structures with deconvolution networks and weak labels: 2121–2125.

Araya-Polo, M., T. Dahlke, C. Frogner, C. Zhang, T. Poggio, and D. Hohl, 2017, Automated fault detection without seismic processing: The Leading Edge, 36, 208–214.

dGB Earth Sciences, 1987, The Netherlands Offshore, The North Sea, F3 Block - Complete: https://bit.ly/2B44GO0.

Di, H., Z. Wang, and G. AlRegib, 2018, in Real-time seismic-image interpretation via deconvolutional neural network: 2051–2055.

Doornenal, J. C., 2014, Kilka uwag o kartografii wglębiowej i modelowaniu geologicznym w holandii: Przegłäd Geologiczny, 62, 806.

Dramsch, J. S., and M. Lthje, 2018, in Deep-learning seismic facies on state-of-the-art CNN architectures: 2036–2040.

Duin, E., J. Doornenal, R. Rijkers, J. Verbeek, and T. E. Wong, 2006, Subsurface structure of the netherlands-results of recent onshore and offshore mapping: Netherlands Journal of Geosciences, 85, 245.

Hochreiter, S., Y. Bengio, P. Frasconi, J. Schmidhuber, et al., 2001, Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Huang, L., X. Dong, and T. E. Clee, 2017, A scalable deep learning platform for identifying geologic features from seismic attributes: The Leading Edge, 36, 249–256.

Long, J., E. Shelhamer, and T. Darrell, 2015, Fully convolutional networks for semantic segmentation: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440.

Mijnlieff, H., 2002, Top pre-permian distribution map and some thematic regional geologic maps of the netherlands: ICCP.

Noh, H., S. Hong, and B. Han, 2015, Learning deconvolution network for semantic segmentation: Proceedings of the IEEE International Conference on Computer Vision, 1520–1528.

Qian, F., M. Yin, X.-Y. Liu, Y.-J. Wang, C. Lu, and G.-M. Hu, 2018, Unsupervised seismic facies analysis via deep convolitional autoencoders: GEOPHYSICS, 83, A39–A43.

Remmelts, G., 1996, Salt tectonics in the southern north sea, the netherlands, in Geology of Gas and Oil under the Netherlands: Springer, 143–158.

Rutherford Ildstad, C., and P. Bormann, 2017, MalenoV: Tool for training and classifying SEGY seismic facies using deep neural networks: Bolgebrygg/MalenoV

Scheck-Wenderoth, M., and J. Lamarche, 2005, Crustal memory and basin evolution in the central european basin systemnew insights from a 3d structural model: Tectonophysics, 397, 143–165.

Schrooßt, B., and H. De Haan, 2003, An improved regional structural model of the upper carboniferous of the cleaver bank high based on 3d seismic interpretation: Geological Society, London, Special Publications, 212, 23–37.

Shafiq, M. A., M. Prabhushankar, H. Di, and G. AlRegib, 2018, in Towards understanding common features between natural and seismic images: 2076–2080.

Shi, Y., X. Wu, and S. Fomel, 2018, in Automatic salt-body classification using deep-convolutional neural network: 1971–1975.

Van Adrichem Bogaert, H., and W. Kouwe, 1993, 1997: Stratigraphic nomenclature of the Netherlands, revision and update. Mededelingen Rijks Geologische Dienst, 50.

Waldeland, A., and A. Solberg, 2017, Salt classification using deep learning: Presented at the 79th EAGE Conference and Exhibition 2017.

Zhao, T., 2018, in Seismic facies classification using different deep convolutional neural networks: 2046–2050.

Zhao, T., V. Jayaram, A. Roy, and K. J. Marfurt, 2015, A comparison of classification techniques for seismic facies recognition: Interpretation, 3, SAE29–SAE58.

Ziegler, P. A., 1988, Evolution of the arctic-north atlantic and the western tetly.

——–, 1990, Geological atlas of western and central europe: Presented at the , Geological Society of London.