Replies to critical reviews

Critical review from team 8:

The content of 9th slide can be reduced. It contains too much word and is not easy to follow during the presentation.

Our response: Thank you so much for your suggestion for the presentation slides formatting. We will be more careful and do better arrangement.

Could you explain more about why choosing densenet121 as your encoder model? Is it because this network achieves better performance than other networks after testing or results from the reference paper?

Our response: In our model, the encoder part was completely trained from nothing and we didn't use the pretrained DenseNet121. You might remember that wrong for our group. For the contracting path in our model, it's the same as the normal convolutional neural network and there is no copy of the output of a single layer passed down to all its subsequent layers, which is entirely different from what DenseNet does.

Critical review from team 57:

With an accuracy of 92%, which is quite high, I would like to have seen an example of one of the 8% instances where the model was mistaken. Perhaps seeing some of those would give insight into how to further improve the model.

Our response: The accuracy is the overall accuracy of all the pixels of all the training images, validation images. It's not an accuracy for some specific image. The final results in our presentation show the differences between the ground truth mask images and our predictions. The different binary pixels are the errors and consist of that 8%.

Critical review from team 77:

As you have explained the traditional method in detail, maybe you can also explain more on your method and model to compare them.

For the model explanation slide, perhaps bullet points are more easy for audiences to follow rather than paragraphs.

For the result, it would be more understandable if you can add several sentences or phrases describing them on the slides.

Our response: Thank you so much for your advice and we will do better arrangement on the presentation slides.

SALT IDENTIFICATION WITH U-NET

Zhuo Chen, Ruihao Wei, and Xiaowen Zhang

University of California San Diego, La Jolla, CA 92093-0238

ABSTRACT

Seismic imaging of salt can help extract oil and gas more safely and efficiently, but it requires human expert interpretation. We propose a convolutional neural network model to identify salt given seismic images. The proposed model is based on U-Net architecture. We implemented the model on TGS Salt Identification Challenge dataset and achieved satisfying results.

Index Terms— CNN, U-Net

1. INTRODUCTION

There are several areas of Earth with huge deposits of salt below the surface, and these areas also contain large accumulation of oil and gas. Seismic imaging of salt bodies can help with efficient extraction of oil and gas, and reduce dangerous situations for oil and gas company drillers. Unfortunately, professional seismic imaging of salt bodies is notoriously difficult and still requires expert human interpretation, since salt bodies have distinctive acoustic features, various compositions and complex shapes [1].

1.1. Problem Description

Given an input seismic image $x_i \in [0; 255]^{m \times n \times 3}$ of integer pixel values in domain X, the corresponding mask image $y_i \in [0; 255]^{m \times n \times 3}$ of integer pixel values in domain Y, where m is the image height, n is the image width, and 3 represents the three RGB channels.

Our goal is to develop a model to learn a mapping function $G_{XY} : X \to Y$, in which the output images $G_{XY}(X)$ is indistinguishable from the ground truth images Y using an cross entropy loss.

1.2. Results Overview

In the project, the input images should be a set of seismic images, and the output should be the corresponding mask images.

2. RELATED WORK

2.1. Traditional Approach

Traditional techniques of salt identification studies main characteristics of texture inside salt structures. Berthelot *et al.* studied three groups of texture attributes: gray-level co-occurrence matrix (GLCM) attributes, frequency-based attributes, and dip and similarity attributes [2]. Various combination of three are used to perform supervised Bayesian classification to find a smooth, continuous border delimiting the salt structure. Amin *et al.* combined the Gray Level Co-occurrence Matrix (GLCM) attributes and the Gradient of Texture (GoT) attributes to conduct dictionary based classification on salt imaging [3]. The algorithm uses a minimum set of features and is immune to the amplitude variations in seismic data. Overall, traditional techniques can achieve high accuracy bu require not only extracting highly representative attributes, but also separately training a classification algorithm based on these attributes, both manually.

2.2. Deep Learning Approach

Recently, geoscientists have started to adopt deep learning techniques in salt imaging. Convolutional neural networks combine attribute extraction and classification, so Wadeland *et al.* applied this technique to salt classification in seismic datasets and showed that it is sufficient to train a classifier on one labelled inline slice to classify other slices in the same dataset [4]. Shi *et al.* designed a multi-layer convolutional neural network capable of automatically capturing subtle salt features, which can be generalized to blind test data [5]. Cruz *et al.* developed an automatic salt segmentation solution based on fully convolutional networks and Transfer learning [6]. Scientist have also developed CNN model that solves automatic interpretation of salt bodies [7]. Even though CNN is proven to be successful in salt identification, it requires many thousand annotated training samples.

2.3. U-Net

Olaf *et al.* introduced U-Net for biomedical image segmentation [8]. It relies on data augmentation, consisting of a contracting path and a symmetrix expanding path, can be trained

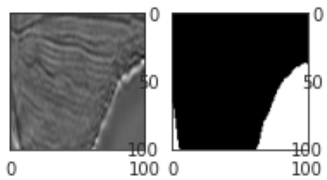

Fig. 1. Left: original seismic image. Right: mask.

end-to-end with a lot less images and shorter runtime. In addition, this neural network performs no worse than the other ones. Therefore, it is widely used for image segmentation problems.

3. DATA

The dataset used is from TGS Salt Identification Challenge in Kaggle. The data consists of a set of images chosen randomly from the subsurface of various locations. There are 4,000 seismic images in the size of $101 \times 101 \times 3$, and corresponding 4,000 binary mask images in the same size. Fig.1 shows an example image from the dataset. The image on the left is the original seismic image and the image on the right is its mask. For the mask image, the white represents salt and black represents all the other ingredients. The unique id and depth of the imaged location are provided for all images.

3.1. Data Preparation and Preprocessing

The data is split into into a training set and validation set with ratio $4 : 1$ (i.e 3200 images in the training set and 800 images in the validation set) with stratification method in section 4.1.

Since the original image size is $101 \times 101 \times 3$, the output size would be odd after it passing through a max pooling layer and the size of the final result can not be maintained the same as the original. To avoid the occurrence of odd image sizes, all images are resized to $128 \times 128 \times 3$ at the beginning.

4. METHODS

4.1. Stratification

It is not efficient to directly use the original salt images and its corresponding mask images. If the entire training set can be divided into different classes based on the coverage rate of the mask image, the training validation split will be based on this classification, which will also generate more scientific result. This step is called stratification. The coverage rate of each mask image is computed as:

$$\rho = \frac{\sum \mathbb{1}}{128 \times 128}$$

Since the mask image is a binary map, the total number of pixels of the salt part would be the entire summation over the mask image. And the coverage rate would be the ration between this count and the entire image size. Based on the coverage rate, the images can be stratified into ten classes by every ten percent of coverage rate.

4.2. General Fully Convolutional Neural Network

Salt identification is intrinsically a problem of classification: determine whether a pixel in the seismic image is salt or not. Generally the classification problem can be solve using fully connected convolutional neural neural network. As Fig. 2 shows, the output of each conv layer is a three-dimensional array of size $H \times W \times D_i$, where H and W is the height and weight of the original image. It can be regarded as doing the number of $H \times W$ classification tasks simultaneously. And D_i is the dimension of the corresponding hidden layer. And the final convolutional layer will give an output of size $H \times W \times C$, where C is the total number of classes needed to be classified which is exactly 2 in salt identification. Then the normalized classification score (probability) for each pixel is calculated by the Softmax function:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_1^C e^{z_i}}$$

The prediction class of pixel p_i is:

$$p_i = argmax_c \sigma(z_i)$$

The main issue of the normal fully convolutional neural network is obvious: the size of the output of each convolutional layer should be consistent with that of the original image. This implies that the number of intermediate variables will be extremely large as well as the computational complexity will be high.

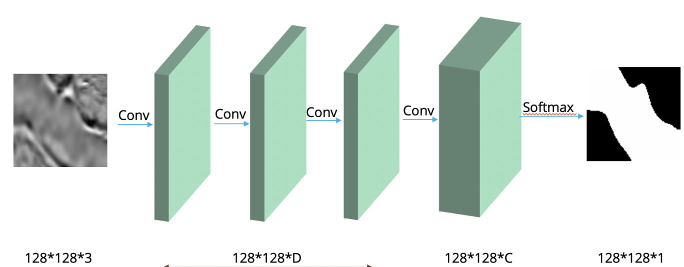

Fig. 2. Architecture of Fully Convolutional Neural Network

4.3. U-net Model

To reduce the consumption of memory and calculation, the model called U-net is used. The architecture is shown in Fig. 3. Basically, the network consists of two parts: the first half is the contracting path and the second half is expansive path. This is very similar to Auto-encoder which consists of encode and decoder. Contracting path will shrinks the size of the output of each convolutional layer by half but increase its depth by two times. Symmetrically, Expansive path will enlarge the size of the output of the convolutional layer while cutting the depth by half. So the final result still maintain the same size as the original image but the total number of variables and the intermediate computation and memory are saved.

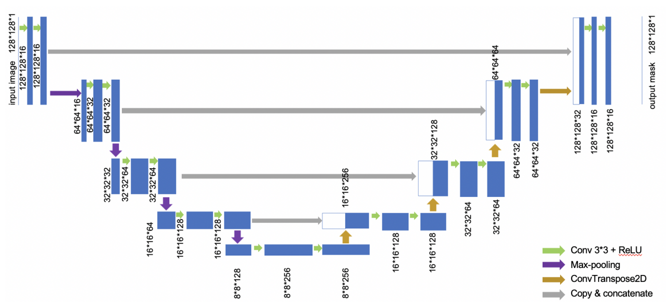

Fig. 3. The U-Net model architecture

To downsample the result of each convolutional layer, the method of max pooling is used. For each sub-block of size $k \times k$ in a three dimensional array, the largest element is maintained so the size is reduced by k times (Fig. 4).

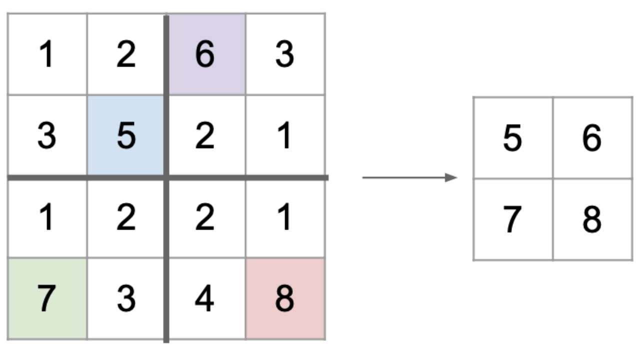

Fig. 4. Example of DownSample

This method is completely the same as what is done in Auto-encoder. However, ConvTranspose neural network layer is used to expand the image instead of normal Upsample layer. Even both of them can expand the image, upsampling layer only uses the nearest neighbor pixels (Fig. 5) while the ConvTranspose is a convolutional operation with learnable parameters. So definitely, the Convtranspose shows better performance and its results is more scientific and reliable (Fig. 6).

The second difference between U-Net and Auto-encoder is that the inputs of the convolutional layers in the expansive path are the concatenation of the deconvolution output and the previous output of the corresponding convolution result in the contracting path. By doing so, we can have better prediction as more information is used.

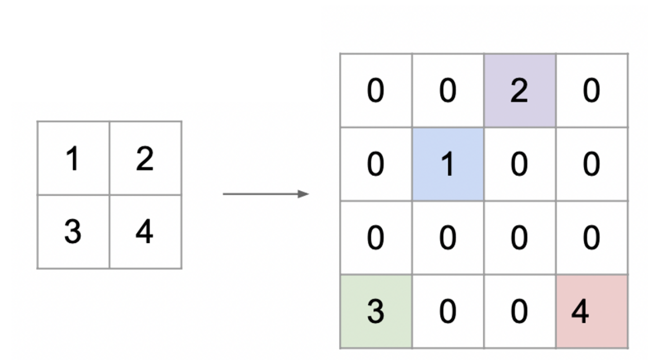

Fig. 5. Example of UpSample

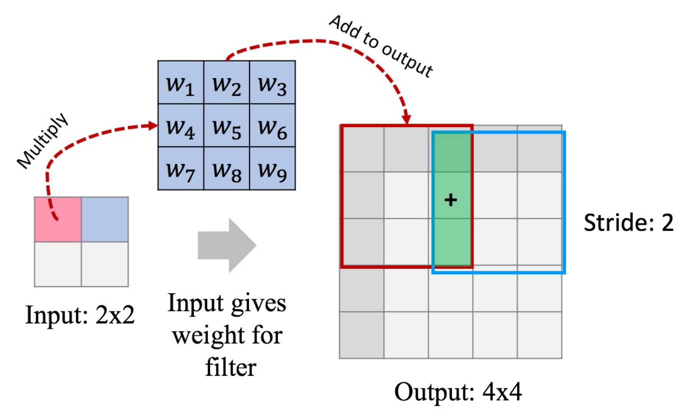

Fig. 6. Example of ConvTranspose

5. EXPERIMENTS AND RESULTS

5.1. Stratification

The coverage rate is compute to divide the training data set into several different classes which can benefit the training and validation split during the training process. The histogram of the coverage rate and corresponding classification distributions are shown in Fig. 7.

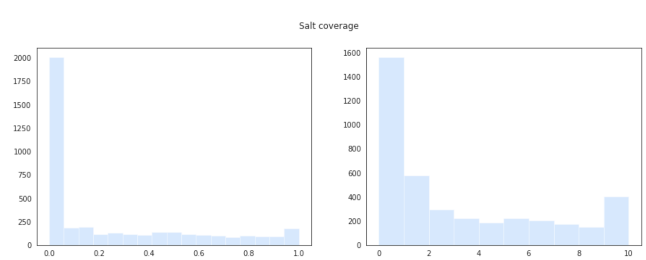

Fig. 7. Salt Coverage Rate And Class Distribution

5.2. Training Process

After several hours of training and tuning parameters, a well-trained model with over 92 percent accuracy is eventually obtained. The Binary Cross Entropy loss and Adam optimizer are used. The Drop out rate is 0.05. The loss curve and the accuracy is shown in Fig. 8 and Fig. 9.

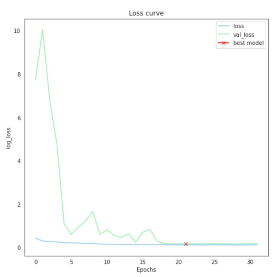

Fig. 8. Loss Curve

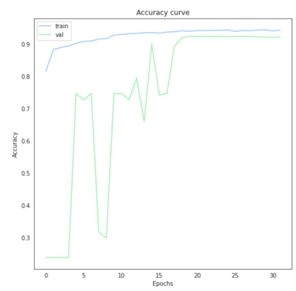

Fig. 9. Accuracy Curve

It can be seen obviously that the losses of training and validation set are decreasing during the training process and the accuracy is increasing.

5.3. Salt Identification Mask

The final identification results of the training set and validation set are shown in Fig. 10 and Fig. 11. The first column is the original seismic image. The second column is the ground truth binary mask. The third column is the predicted probability of each pixel. The fourth column is the predicted binary mask.

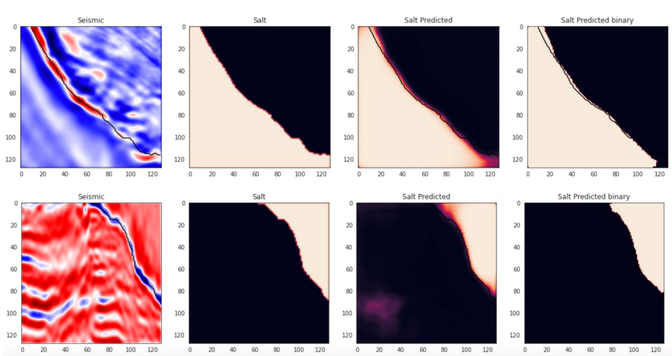

Fig. 10. Identification Result of Training Set

As you can see, there are only slight differences between the ground truth and out prediction for the training set (Fig. 10). But for the validation and test set, there would be huge differences (Fig. 11).

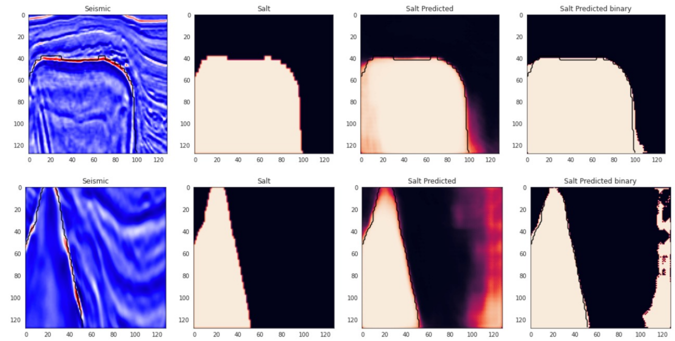

Fig. 11. Identification Result of Validation Set

6. CONCLUSION AND FUTURE WORK

Basically, we achieved the goal of salt identification task with good results. Furthermore, we have learned more knowledge about convolutional neural network especially the U-Net model and obtained more practice on building and training neural network in this project. And we also accumulate experience of solve practical problems using machine learning as an efficient tool.

If we have more time, we would try several new things. First, we would like to use Residual Networks (ResNet) build the segmentation model. Since ResNet helps to solve the degradation problem that saturates the accuracy of the model, we would expect to see an accuracy improvement of our model with the use of ResNet. Second, we also would like to try a variation of the U-net architecture called attention U-net instead of the standard one that we implemented in this project. We would like to add attention module to perform class-specific pooling, which we expect to result in a more accurate and robust image classification performance

7. CONTRIBUTION

Ruihao is in charge of literature review of traditional techniques and parameter tuning; Xiaowen is in charge of literature review about deep learning, data preparation and preprocessing; Zhuo is in charge of training U-Net and testing data.

8. REFERENCES

[1] Ian F. Jones and Ian Davison. Seismic imaging in and around salt bodies. *Interpretation*, 2(4):SL1–SL20, 2014.

[2] Angélique Berthelot, Anne H.S. Solberg, and Leiv-J. Gelius. Texture attributes for detection of salt. *Journal of Applied Geophysics*, 88:52 – 69, 2013.

[3] Asjad Amin*, Mohamed Deriche, Tamir Hegazy, Zhen Wang, and Ghassan AlRegib. *A Novel Approach for Salt Dome Detection using A Dictionary-based Classifier*, pages 1816–1820. 2015.

[4] Anders U. Waldeland and Anne H. S. Solberg. Salt classification using deep learning. 2017(1):1–5, 2017.

[5] Yunzhi Shi, Xinming Wu, and Sergey Fomel. *Automatic salt-body classification using deep-convolutional neural network*, pages 1971–1975. 2018.

[6] P. M. Cruz and J. P. Navarro. Salt segmentation with fully convolutional networks and transfer learning. 2018(1):1–5, 2018.

[7] Anders U. Waldeland, Are C. Jensen, Leiv-J. Gelius, and Anne H. S. Solberg. Convolutional neural networks for automated seismic interpretation. *The Leading Edge*, 37(7):529–537, 2018.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.