Salt Detection Using Segmentation of Seismic Image

Mrinmoy Sarkar
Department of Electrical & Computer Engineering
North Carolina A&T State University
Greensboro, NC-27411, USA
msarkar@aggies.ncat.edu

***Abstract*—In this project, we present state-of-the-art deep convolution neural network (DCNN) to segment seismic image for salt detection below the earth surface. Detection of salt location is very important for starting mining. Hence, seismic image is used to detect the exact salt location under the earth surface. However, precisely detecting the exact location of salt deposits is very difficult. Therefore, professional seismic imaging still requires expert human interpretation of salt bodies. This leads to very subjective, highly variable renderings. Hence, to create the most accurate seismic images and 3D renderings, we need a robust algorithm that automatically and accurately identifies if a surface target is salt or not. Since, the performance of DCNN is well-known and well-stablished for object recognition in image, DCNN is a very good choice for this particular problem. We successfully applied DCNN to a dataset of seismic images in which each pixel is labeled as salt or not. The result of this algorithm is promissing.**

***Index Terms*—Seismic Image, Image Segmentation, DCNN, Auto-Encoder**

OBJECTIVES

- Segmentation of seismic image into salt or sediment using DCNN.
- Automate the process of analysis of seismic image.
- Reduce the cost of identifying an earth surface before mining.

I. INTRODUCTION

A seismic image is produced from imaging the reflection coming from rock boundaries. The seismic image shows the boundaries between different rock types. In theory, the strength of reflection is directly proportional to the difference in the physical properties on either side of the interface. While seismic images show rock boundaries, they don't say much about the rock themselves; some rocks are easy to identify while some are difficult. There are several areas of the world where there are vast quantities of salt in the subsurface. One of the challenges of seismic imaging is to identify the part of subsurface which is salt. However, it is a image segmentation problem from the image processing perspective. There are many robust algorithms available for this task in the literature such as feature-space, image-domain and physics based techniques [1]. These techniques have been successfully used for color image segmentation captured from digital camera. Since seismic images are significantly different than digital images, those state-of-the-art techniques fail for segmentation task. There are many challenges for seismic image segmentation. some of them are listed as follows:

- Image capturing method.
- Uneven distribution of salt and other rocks.
- Rock which have density compared to salt.
- Only gray-level image means lack of information.
- Uneven structure of rocks below the earth surface which causes uneven reflection.

However, there are many state-of-the-art machine learning technique that can be used to solve this problem. The most promissing technique in the literature is deep convolution neural network for any task related to image. This technique has been used for object recognition [2], image segmentation [3], style transformation [4], human action recognition [5], medical image segmentation [6] and image denoising [7]. Hence, we have used this method to solve the problem at hand. In this work, our contributions are as follows:

- We used state-of-the-art DCNN to segment seismic image for salt identification.
- We automated the post analysis of seismic images.
- We reduced the cost for seismic image analysis.
- We relaxed the necessity of human expert for seismic image segmantation.

The rest of the paper is organised as in section II literature survey, in section III our method to solve the segmentation problem, in section IV experimental results of our method and concludes with conclusion & future work in section V.

II. LITERATURE SURVEY

Image segmentation is a fundamental task for many image processing, video analysis or computer vision application. Hence, many research papers have been published on this topic. All the proposed method can be catagorised into the following three techniques [1].

1) Feature-Space Based Techniques
2) Image-Domain Based Techniques
3) Physics Based Techniques

A. Feature-Space Based Techniques

In this approach, color is assumed to be a constant property of the surface of each object within an image. So, every pixel can be clustered or grouped into some region within the

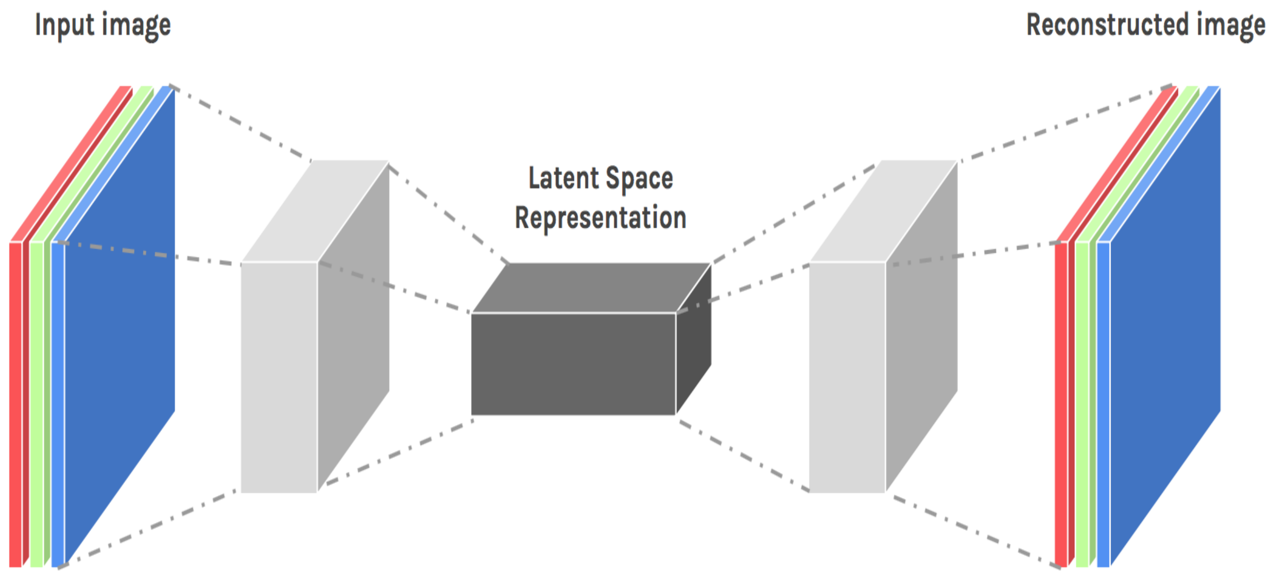

Fig. 1. Convolutional Autoencoder architecture. [8]

image, which will produce the segmented image. Clustering and histogram thresholding are two well-known feature-space based techniques [1].

B. Image-Domain Based Techniques

The feature space based algorithms work on global property of the image which satisfy the homogenity requirement of image segmentation. However, those techniques do not consider the spation characteristic of the image. Thue researchers found image domain based technique. These techniques satisfy both feature-space homogeneity and spatial compactness at the same time. The spatial compactness is ensured either by subdividing and merging or by progressively growing image regions, while the homogeneity is adopted as a criterion to direct these two processes. According to the strategy preferred for spatial grouping, these algorithms are usually divided into split-and-merge and region growing techniques. Neural-network based classification, split and merge using region adjacency graph(RAG) and edge based algorithmss are known to be image domain based techniques [1].

C. Physics Based Techniques

The discussed techniques are prone to segmentation error when the image is affected by highlights, shadowing and shadows. The problem can be solved considering the interaction of light with colored materials and to introduce models of this physical interaction in the segmentation algorithms. This is the reason these techniques are known as physics based techniques. The mathematical tools used in these techniques are quite simillar to the previous two types of technique; the major difference with respect to those is the underlying physical model developed for the reflections properties of colored matter [1].

As we are using seismic image, the images are not affected by shadowing or shawds. Our techniques lies in the image domain based techniques. However, the architechture and methodology is completely different than the techniques discussed in [1].

III. Methodology

In this project, we have used autoencoder(AE) architecture of neural network. Autoencoders (AE) are a family of neural networks for which the input is the same as the output. They work by compressing the input into a latent-space representation, and then reconstructing the output from this representation. If the architecture is built up on convolution neural networkthen it is known as Convolutional Autoencoder (CAE). Since out input is image, we used CAE to develop our segmentation model. The CAE architecture is shown in Fig. 1. There are two parts of this architecture, named as encoder and decoder. The encoder part consist of convolution and pooling layers and the decoder part consists of convolution and upsampling layers. Each of these layers are describes in the following section.

A. Convolution Layer

The convolution layer is the main building block of CNN architecture. The primary purpose of convolution in case of a CNN is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. In this layer the input image is convolved with some predefined filters or kernel and then the output of the convolution operation is fed to an activation function. The $2D$ convolution operation is

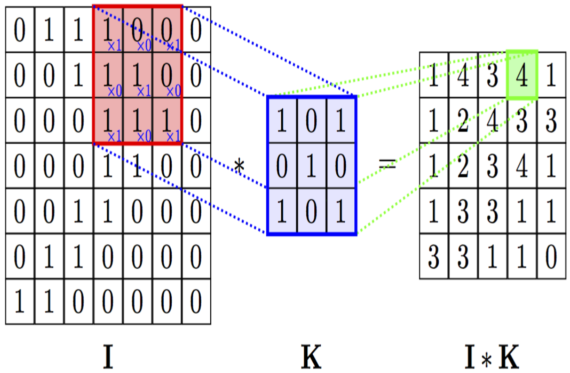

Fig. 2. Convolution operation. [9]

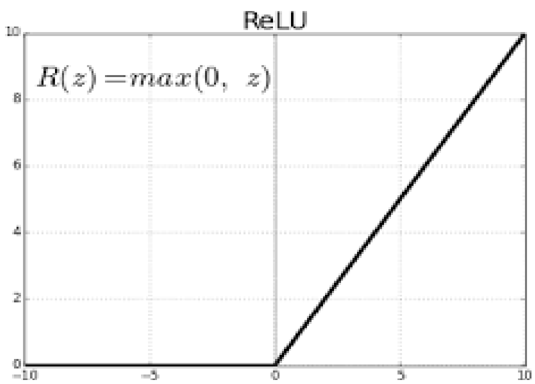

Fig. 3. ReLU activation function. [10]

shown in Eqn. 1. The output of the convolution operation is known as Feature Map. The size of the Feature Map is controlled by three parameters: [11]

a) Depth: Depth corresponds to the number of filters used for the convolution operation.

b) Stride: Stride is the number of pixels by which the filter matrix is slided over the input matrix.

c) Zero-padding: Sometime zeros are padded around the border so that the filters can be applied to the bordering elements.

The convolution operation is shown in Fig. 2. The last operation of a convolution layer is activation function. The most common activation function for CNN is rectified linear unit function (ReLU). The ReLU function is shown in Fig. 3.

$$f(m,n) \circledast g(m,n) = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} f(i,j) \times g(m-i, n-j) \quad (1)$$

B. Pooling Layer

A pooling layer is another building block of a CNN. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates on each feature map independently. The most common approach used in pooling is max pooling. The max pooling operation is shown in Fig. 4.

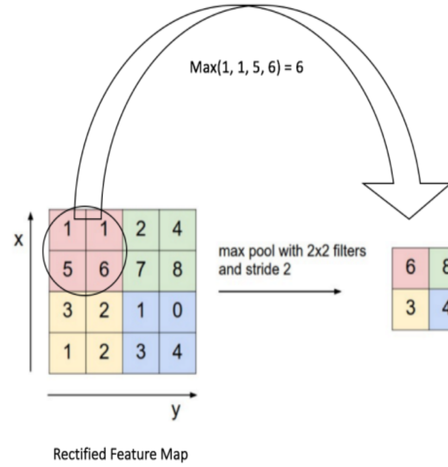

Fig. 4. Operation in pooling layer. [11]

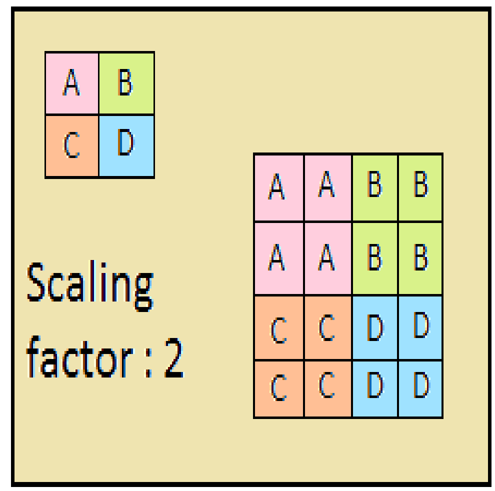

Fig. 5. Image resizing (Nearest-Neighbor method). [12]

C. Upsampling Layers

In this layer the input image is resized to a higher dimention. There are different methods for resizing the image from lower dimention to higher dimention. One of those algorithms is nearest-neighbor algorithm. In this algorithm, the nearest pixels are copied from the original pixel value. For a scaling factor of 2 the output of nearest neighbor algorithm is shown in Fig. 5.

D. Architecture of the Developed Network

The developed architecture is a combination of the three layers described above sections. The details of the architecture is given in Table I.

E. Loss function

To train the network, reduce mean of sigmoid cross entropy is used as the loss function. If x is the predicted label and z

TABLE I
ARCHITECTURE OF THE DEVELOPED AUTOENCODER MODEL

				Encoder			
	layer1	layer2	layer3	layer4	layer5	layer6	layer7
Name	conv2d	max_pooling2d	conv2d	max_pooling2d	conv2d	max_pooling2d	conv2d
No. of Filter	64		64		32		32
Filter dimension	3x3		3x3		3x3		3x3
Activation function	relu		relu		relu		relu
Pool size		2x2		2x2		2x2	
Stride size		2x2		2x2		2x2	
	layer8	layer9	layer10				
Name	max_pooling2d	conv2d	max_pooling2d				
No. of Filter		16					
Filter dimension		3x3					
Activation function		relu					
Pool size	2x2		2x2				
Stride size	2x2		2x2				
				Decoder			
	layer11	layer12	layer13	layer14	layer15	layer16	layer17
Name	upsampler	conv2d	upsampler	conv2d	upsampler	conv2d	upsampler
No. of Filter		16		32		32	
Filter dimension		3x3		3x3		3x3	
Activation function		relu		relu		relu	
Output Image size	8x8		16x16		32x32		64x64
Resize method	Nearest Neighbor		Nearest Neighbor		Nearest Neighbor		Nearest Neighbor
	layer18	layer19	layer20	layer21	layer22	layer23	
Name	conv2d	upsampler	conv2d	downsampler	conv2d	output	
No. of Filter	64		64		1		
Filter dimension	3x3		3x3		3x3		
Activation function	relu		relu			sigmoid	
Output Image size		128x128		101x101			
Resize method		Nearest Neighbor		Nearest Neighbor			

is the true label then the sigmoid cross entropy can be written as equation 2 and if z is a set of n different values and m is the total number of training samples then the reduce mean loss can be calculated using equation 4.

$$\begin{aligned} &sigmoid\_cross\_entropy(x, z) \\ &= z \times (-\log(sigmoid(x))) \\ &+ (1-z) * (-\log(1 - sigmoid(x))) \end{aligned} \tag{2}$$

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{3}$$

$$Loss = \frac{1}{m} \times \sum_{j=1}^{m} \sum_{i=1}^{n} sigmoid\_cross\_entropy_j(x_i, z_i) \tag{4}$$

F. Training Algorithm

The training algorithm used for optimizing the pre-defined loss function is ADADELTA. As describes in [13], the ADADELTA is an adaptive gradient descent algorithm which adapts the learning rate dynamically during the training process based on first order information only and the computational cost of the algorithm is less than any other state-of-the-art gradient descent algorithm. For more information about this optimization technique, readers can look into [13].

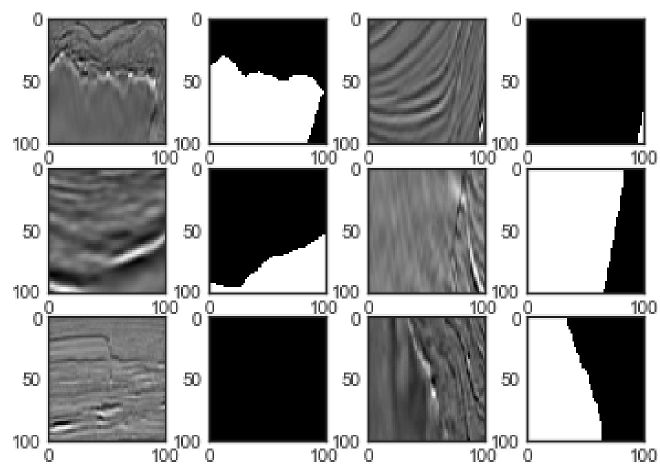

Fig. 6. Sample seismic images and corresponding mask images.

IV. EXPERIMENTAL RESULTS

The dataset used for this project, contains 4000 seismic images with 4000 labeled mask images. The seismic images are in gray scale and the mask images are in black and white. White pixel in mask image indicated the presence of salt in original image. The size of all the images are 101×101. The input images are resized to 128×128 for the sake of fast computation but the mask images are keep unchanged. Some sample seismic and mask images are shown in Fig. 6. The data set is obtained from [14]. Software tools used for this project, are listed below:

1) Python
2) scikit-learn
3) TensorFlow
4) Keras
5) Pandas
6) Numpy
7) Matplotlib

With an initial learning rate of 0.001 and with a mini batch size of 100, after 5000 epoches the training loss was 0.3297 and the test loss was 0.3464. However, after 10000 epoches the training loss was 0.2406 and the test loss is 0.2666.

V. Conclusion & Future Work

References

[1] L. Lucchese and S. K. Mitra, "Colour image segmentation: a state-of-the-art survey," *Proceedings-Indian National Science Academy Part A*, vol. 67, no. 2, pp. 207–222, 2001.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[4] S. A. Etemad and A. Arya, "3d human action recognition and style transformation using resilient backpropagation neural networks," in *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, vol. 4. IEEE, 2009, pp. 296–301.

[5] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[6] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 565–571.

[7] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[8] "Autoencoders - deep learning bits," https://hackernoon.com/autoencoders-deep-learning-bits-1-11731e200694.

[9] "2d convolution," https://github.com/PetarV-/TikZ/tree/master/2D%20Convolution.

[10] "Relu : Not a differentiable function: Why used in gradient based optimization? and other generalizations of relu." https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec.

[11] "An intuitive explanation of convolutional neural networks," https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/.

[12] "Image scaling methods and matlab implementations," https://aditharajakaruna.wordpress.com/2013/07/12/image-scaling-methods-and-matlab-implementations/.

[13] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[14] "Tgs salt identification challenge," https://www.kaggle.com/c/tgs-salt-identification-challenge.