**A  Summer Internship Report**

**On**

**SEM IMAGE SEGMENTATION**

Submitted by

**Murari Jha**

**Roll No.: 16CH30038**

Under the guidance of

**Prof. Somenath Gaungly**

****

**CHEMICAL ENGINERRING DEPARTMENT**

**INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR**

ABSTRACT

Texture-based feature extraction and object segmentation are challenging in image processing. In this study, the U-Net architecture developed for biomedical image analysis was used to evaluate porous nature of NiO sample within scanning electron microscope (SEM) images. The Pytorch neural network library was used to create semantic segmentation and feature extraction models in differentiating pores from solid. For the work, we prepared 2880 randomly sliced image cuts (128 \* 128 pixels) from three masked image tiles (1024 \* 1024 pixels) with tagged feature objects, among which 2448 are for training and the remaining 432 held image slices for validation. In the validation, Dice Sorensen Coefficient reaches 97.9%. The trained model approved by validation was used for pores segmentation. This approach demonstrates that texture-based deep learning feature extraction is feasible, cost-effective and timely, and can help scientists gain new insights by quantitatively analyzing specific natures and insights from segmentation.

CONTENT

|  |  |  |
| --- | --- | --- |
| 1 | ABSTRACT | 2 |
| 2 | CONTENT | 3 |
| 3 | INTRODUCTION | 4 |
| 4 | DATASET | 5-7 |
| 5 | MODEL | 8-11 |
| 6 | TRAINING MODEL PREPARATION | 12-13 |
| 7 | RESULT | 14-17 |
| 8 | CONCLUSION | 18 |
| 9 | REFERENCES | 18 |

INTRODUCTION

Image segmentation has been an active research field in the past decades. Deep learning approaches play an increasingly important role and has become state-of-the-art in various segmentation tasks [1]. Ultimately, it is desirable to have a fully automatic image processing pipeline including segmentation. It can provide improved reproducibility and more objective simulation outcomes due to excluding subjective assessments of an operator [2]. Modern machine learning (ML) based image segmentation methods look like a perfect candidate for the role. Nowadays, ML and especially deep learning approaches are experiencing explosive growth. The main advantage of a deep learning is the ability to learn complex patterns and features from the data itself. Segmentation by means of Convolutional Neural Networks (CNN) provides state-of-the-art results for many tasks [3]. CNN is one of the most promising approaches for automatic segmentation of SEM images.

Deep learning using Convolutional Neural Network (CNN) is a recently developed approach and represents the state of the art in more complicated image segmentation (LeCun et al., 2015; Falk and Mai, 2019). Deep learning network architectures that are comprised of multiple layers of artificial neural net units—as opposed to traditional task-specific algorithms—allow computational models to learn representations of data. Trained computer models efficiently perform tasks that, by convention, are done by human experts with years of experience in specialized fields. Deep learning techniques have been successfully applied to many fields, such as computer vision, voice recognition, medical image analysis and many other domains including geosciences and geophysics, with results comparable to and in some cases superior to human experts alone (e.g., Aprile et al., 2014; Ciresan et al., 2012; Falk and Mai, 2019; Karpatne et al., 2018; LeCun et al., 2015; Noh et al., 2015; Ronneberger et al., 2015; Tian and Daigle, 2018; Zhang et al., 2015). Deep learning systems are designed and developed for performing specialized tasks, which learn the features relevant for the task from data and use optimization algorithms that are relevant to features of the subjects being studied;

Our aim is to train CNN-based models once and to apply these models many times in the automatic mode for the segmentation of various porous samples. One of the key challenges for the use of supervised machine learning for segmentation of porous surface is the lack of ground truth (GT). Manual pixel-perfect labelling of a large enough 2D dataset would be way too time-consuming, while utilizing significantly higher-resolution images, such as Scanning Electron Microscopy (SEM), possesses its own challenges [4-5]. Next, we present in detail our approach for generation of training and validation datasets, U-net for segmentation. It presents the experimental results for deep neural networks under investigation as well as comparison with conventional segmentation methods like different thresholding method. Finally, the conclusions and discussions of further research directions are contained.

DATASET

The object of this study is to separate void from solid part within porous surface in secondary electron SEM images as shown in Fig. 1 through image semantic segmentation by deep-learning. The field emission SEM images were taken at the XRD lab in the chemical engineering department of Indian Institue of Tecnology Kharagpur. In order to reveal the natural pore character of the surface, no conductive coating was applied, thus low voltage and current were used to avoid charging issue during imaging. The operating parameters for SEM used for imaging were 5 kV and 100nm JEOL. A tile is a single raw standard-sized image (1024\*1024 pixels) from a core sample. A map is a large mosaic image from the same core sample, consisting of multiple standard raw image tiles. An image slice is an image cut from the raw image tile for training and model validation purposes (128\*128 pixels).The Ground truth image is generated by Adaptive Thresholding method.

In **adaptive threshold** unlike fixed threshold, the threshold value at each pixel location depends on the neighbouring pixel intensities. To calculate the threshold *T(x, y)* i.e. the threshold value at pixel location *(x, y)* in the image, we perform the following steps -

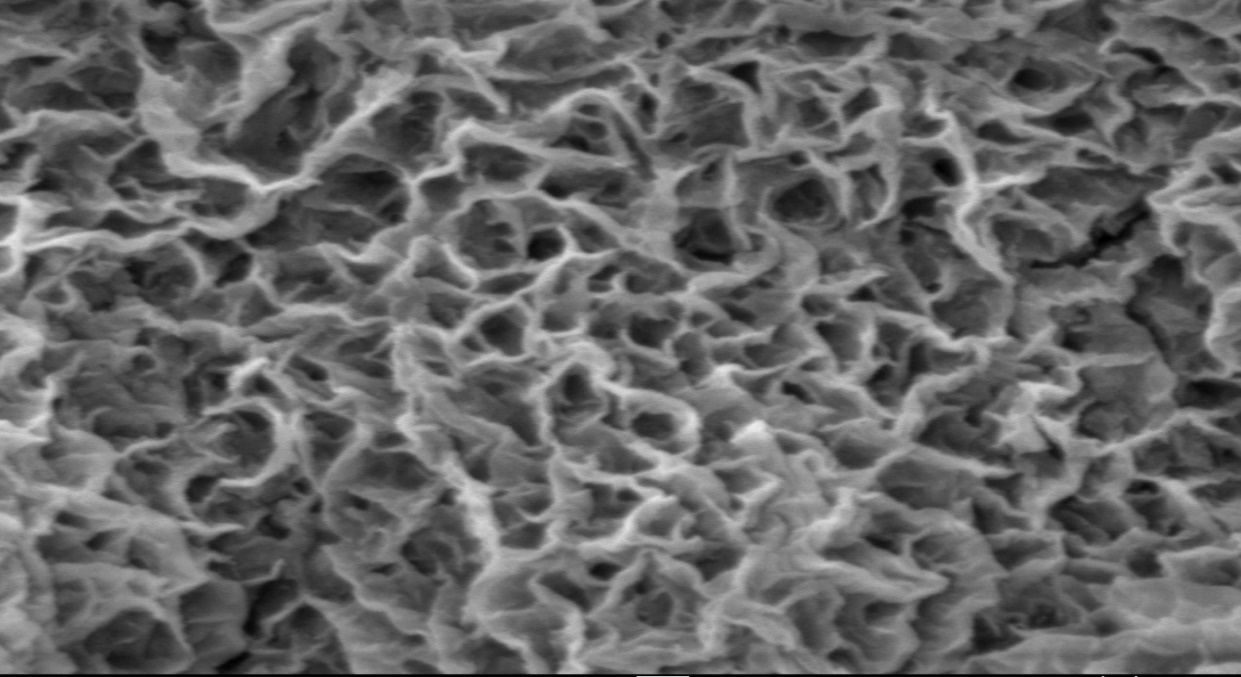
1. A (b x b) region around the pixel location is selected. b is selected by the user.
2. The next step is to calculate the weighted average of the (b x b) region. OpenCV provides 2 methods to calculate this weighted average. We can either use the average (mean) of all the pixel location that lie in the (b x b) box or we can use a Gaussian weighted average of the pixel values that lie in the box. In the latter case, the pixel values that are near to the centre of the box, will have higher weight. We will represent this value by *WA(x, y)*.
3. The next step is to find the Threshold value *T(x, y)* by subtracting a constant parameter, let’s name it param1 from the weighted average value *WA(x, y)* calculated for each pixel in the previous step. The threshold value T(x, y) at pixel location *(x, y)* is then calculated using the formula given below -

*T(x, y) = WA(x, y) - param1*

This is how we get our threshold value. We took b=21 as region and Gaussian for weighted average and param1= 2

|  |  |
| --- | --- |
| Category | Number Of Images |
| Total Images | 2880 |
| Training | 2448 |
| Validation | 432 |

**Table 1**. Dataset Statistics



**Fig 1**. SEM Image for training the model

Training Data Examples

Input Images

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 101 | 201 | 301 |
| 401 | 1001 | 1201 | 2000 |

Ground Truth

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 101 | 201 | 301 |
| 501 | 1001 | 1201 | 2000 |

MODEL

**BASELINE**

**Li Thresholding:** In 1993, Li and Lee proposed a new criterion for finding the “optimal” threshold to distinguish between the background and foreground of an image. They proposed that minimizing the cross-entropy between the foreground and the foreground mean, and the background and the background mean, would give the best threshold in most situations.

Li’s iterative method uses gradient descent to find the optimal threshold. If the image intensity histogram contains more than two modes (peaks), the gradient descent could get stuck in a local optimum. An initial guess for the iteration can help the algorithm find the globally-optimal threshold.

**APPROACH**

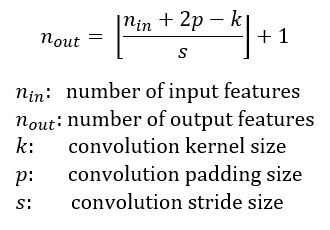
Before we dive into the UNET model, it is very important to understand the different operations that are typically used in a Convolutional Network.

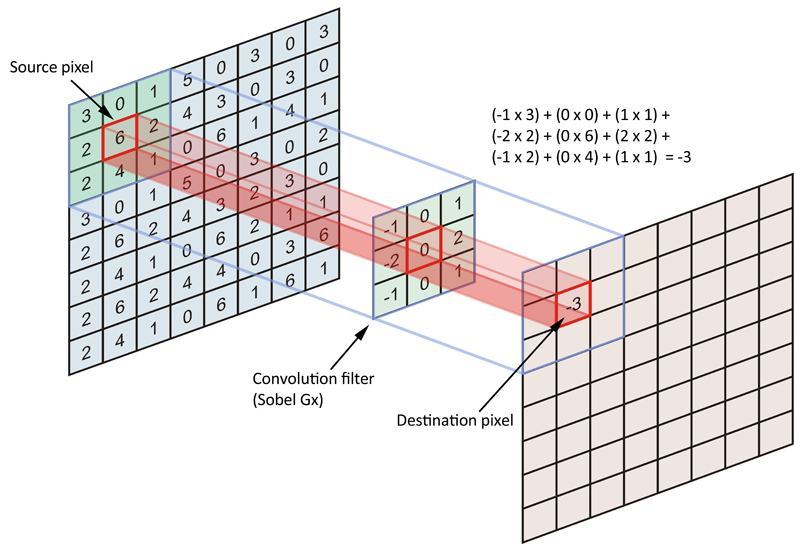
**(i). Convolution operation**

There are two inputs to a convolutional operation.

1. A 3D volume (input image) of size (nin x nin x channels).
2. A set of ‘k’ filters (also called as kernels or feature extractors) each one of size (f x f x channels), where f is typically 3 or 5.

The output of a convolutional operation is also a 3D volume (also called as output image or feature map) of size (nout x nout x k). The relationship between nin and nout is as follows:

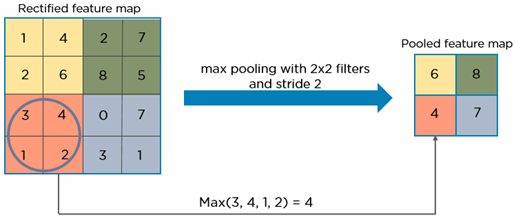




## ii) Max pooling operation

In simple words, the function of pooling is to reduce the size of the feature map so that we have fewer parameters in the network.

For example:



Basically from every 2x2 block of the input feature map, we select the maximum pixel value and thus obtain a pooled feature map. Note that the size of the filter and strides are two important hyper-parameters in the max pooling operation.

## iii) Transposed Convolution

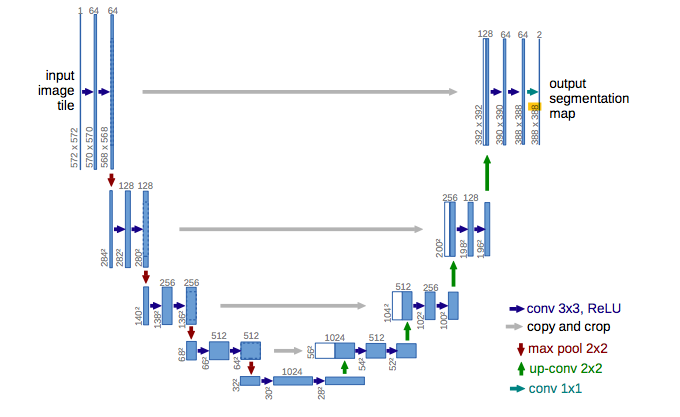
Transposed convolution (sometimes also called as de-convolution or fractionally stride convolution) is a technique to perform up sampling of an image with learnable parameters.

However, on a high level, transposed convolution is exactly the opposite process of a normal convolution i.e., the input volume is a low resolution image and the output volume is a high resolution image.

For this task we used the U-Net Architecture .The [U-Net](https://arxiv.org/abs/1505.04597)was developed by Olaf Ronneberger et al [6]. for Bio Medical Image Segmentation. The architecture contains two paths. First path is the contraction path (also called as the encoder) which is used to capture the context in the image. The encoder is just a traditional stack of convolutional and max pooling layers. The second path is the symmetric expanding path (also called as the decoder) which is used to enable precise localization using transposed convolutions. Thus it is an end-to-end fully convolutional network (FCN), i.e. it only contains Convolutional layers and does not contain any dense layer because of which it can accept image of any size.

The U-net architecture is synonymous with an encoder-decoder architecture. Essentially, it is a deep-learning framework it comprises two parts:

1. A contracting path similar to an encoder, to **capture context via a compact feature map.**
2. A symmetric expanding path similar to a decoder, which allows **precise localization**. This step is done to retain boundary information (spatial information) despite down sampling and max-pooling performed in the encoder stage.



**Fig 2**. U-Net Model Architecture for segmentation

**Advantages of Using U-Net**

1. Computationally efficient
2. Trainable with a small data-set
3. Trained end-to-end
4. Preferable for bio-medical applications

TRAINING MODEL PREPARATION

The training was implemented using 2880 data point. We used 85% of the data for training and the remaining data for validation. We scaled the range of the data from 0.0–1.0. We used the stochastic gradient descent (SGD) optimization method to train the proposed network, and a 128 × 128 pixel patch was chosen sequentially from the input image. The training patches were augmented by randomly flipping them horizontally and vertically and rotating at [30, 45, 60 and 90] degree angle. We used 16 batches for training. The learning rate was initialised as 0.005 and decreased by checking the metrics quantity. The Loss function used is Binary Cross-entropy combining with sigmoid layer. Hyper parameters for training are shown in Table 1.

**Terms in Table [3]:**

**Batch Size**: Batch size is a term used in machine learning and refers to the **number of training examples utilized in one iteration**. The batch size can be one of three options: mini-batch mode: where the batch size is greater than one but less than the total dataset size. Usually, a number that can be divided into the total dataset size.

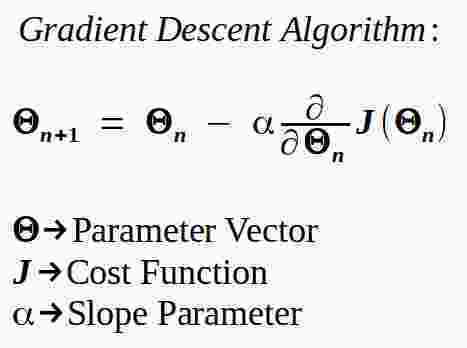
**Learning Rate**: **Learning rate is a hyper-parameter** that controls how much we are adjusting the weights of our network with respect the loss gradient. The lower the value, the slower we travel along the downward slope.

**Loss Function**: a **loss function** or **cost function** is a function that maps an [event](https://en.wikipedia.org/wiki/Event_(probability_theory)) or values of one or more variables onto a [real number](https://en.wikipedia.org/wiki/Real_number) intuitively representing some "cost" associated with the event. An [optimization problem](https://en.wikipedia.org/wiki/Optimization_problem) seeks to minimize a loss function.

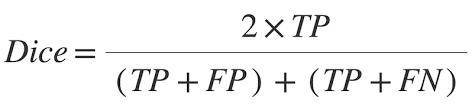
**Bcewithlogitsloss** function combines sigmoid and bceloss functions, that is, before the input passes through bceloss function, it will be converted into probability form through sigmoid function.

ℓ(x,y)=L={l1​,…,lN​}⊤,ln​=−wn​[yn​⋅logσ(xn​)+(1−yn​)⋅log(1−σ(xn​))]

**Stochastic gradient descent** (often abbreviated **SGD**) is an [iterative method](https://en.wikipedia.org/wiki/Iterative_method) for [optimizing](https://en.wikipedia.org/wiki/Mathematical_optimization) an [objective function](https://en.wikipedia.org/wiki/Objective_function) with suitable [smoothness](https://en.wikipedia.org/wiki/Smoothness) properties. It performs a parameter update for *each* training example xi and label yi



**Dice similarity coefficient (DSC)**: It was used as a statistical validation metric to evaluate the performance of segmentations and the spatial overlap accuracy of automated probabilistic fractional segmentation of SEM images.Evaluation metrics of Dice coefficient is used for similarity between result and the ground truth image.

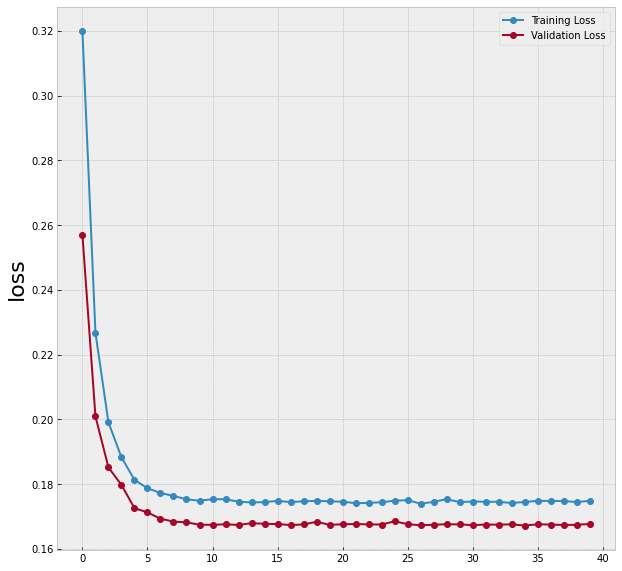




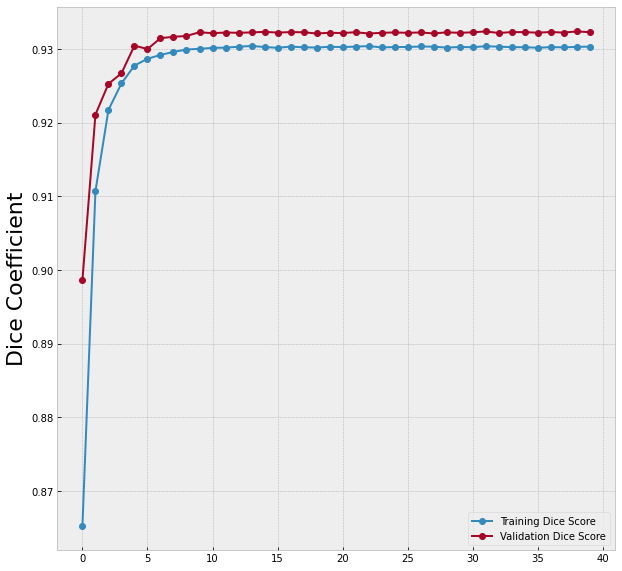
|  |  |
| --- | --- |
| Training Environment | Specification |
| Patch size | 128 × 128 |
| Mini Batch size | 16 |
| Learning Rate | 0.005 |
| Loss Function | BCEWithLogitsLoss |
| Optimizer | SGD |
| Evaluation Criteria | Dice Coefficient |

**Table 2**. Hyper parameters for training model

RESULTS



**Fig 3**. Plot for loss with Epochs during training and validation

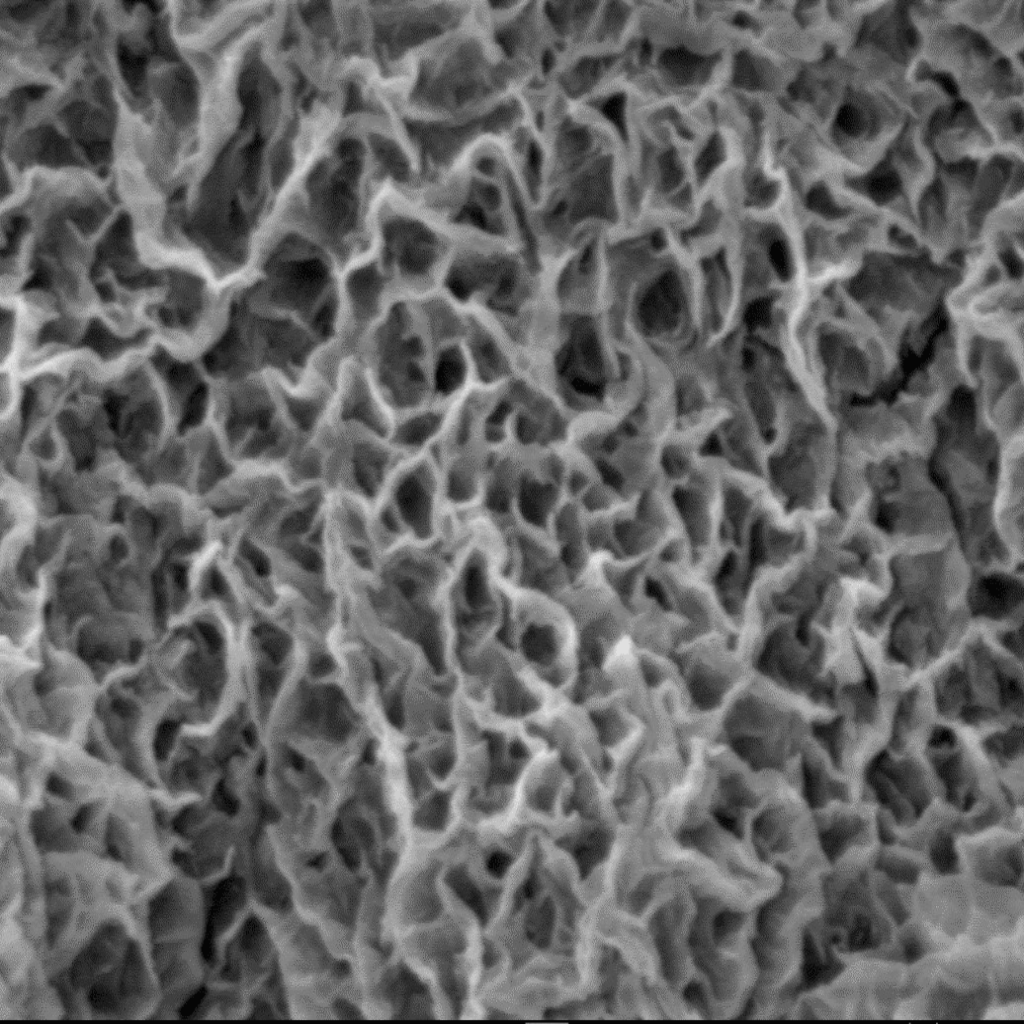


**Fig 4**. Plot for Dice score with Epochs during training and validation

|  |  |
| --- | --- |
| Model | Dice-coefficient |
| Li thresholding | 0.713 |
| U-Net | 0.979 |

**Table 3**. Dice score for Different Model Used

Test Result



**Fig 5**. SEM Image for testing the model

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

(a)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

(b)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**Fig 6**. (a) Input Images for training model, (b) Ground truth Images and (c) Predicted Image from U-Net Model.

|  |  |  |  |
| --- | --- | --- | --- |
| Image Number | Dice-Coefficient in U-Net result | Porosity in U-Net predicted Image | Porosity in Ground Truth Image |
| 1 | 0.973362055 | 0.575866699 | 0.58581543 |
| 2 | 0.976861732 | 0.538085938 | 0.548706055 |
| 3 | 0.979186992 | 0.56085205 | 0.565246582 |
| 4 | 0.978707015 | 0.553649902 | 0.555664063 |
| 5 | 0.976988547 | 0.58001709 | 0.581726074 |
| 6 | 0.976722289 | 0.58404541 | 0.582763672 |
| 7 | 0.977114967 | 0.559631348 | 0.565856934 |
| 8 | 0.978583893 | 0.575744629 | 0.578491211 |

**Table 4**. Porosity and Dice score for Ground truth and predicted Image

CONCLUSION

In this paper, we investigated U-Net based architectures for pixel wise semantic segmentation for void and solid classification. We perform our experiments on SEM images of NiO porous surface generated and ground truth generated by adaptive thresholding method. We found our proposed architectures to outperform the existing baselines significantly. In the future, we would like to explore segmentation on 3D images in porous substance.

REFERENCE

1. LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.<https://doi.org/10.1038/nature14539.PMID26017442>.
2. Zhuoheng Chen a,\*, Xiaojun Liu a, Jijin Yang b, Edward Little a, Yu Zho Deep learning-based method for SEM image segmentation in mineral characterization, an examplefrom Duvernay Shale samples in Western Canada Sedimentary Basin. Elsevier Computers & Geosciences 138 (2020) 104450.
3. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Martinez-Gonzalez, P.; Garcia-Rodriguez, J. A survey on deep learning techniques for image and video semantic segmentation. Appl. Soft Comput. **2018**, 70, 41–65.
4. Varfolomeev, I.; Yakimchuk, I.; Sharchilev, B. Segmentation of 3D image of a rock sample supervised by 2D mineralogical image. In Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 346–350.
5. Varfolomeev, I.; Yakimchuk, I. Segmentation of 3D X-ray microtomographic images of rock samples, supervised by 2D mineral maps. In Proceedings of the Technical Vision in Control Systems (TVCS), Moscow, Russia, 2–14 March 2019; pp. 31–32.
6. Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer- Assisted Intervention (MICCAI), vol. 9351. Springer, LNCS, pp. 234–241.