Proceedings of 14th IOE Graduate Conference
Peer Reviewed
Year: 2023 Month: December Volume: 14
ISSN: 2350-8914 (Online), 2350-8906 (Print)

Study of Semantic Segmentation of Microscopy Images of Activated Carbon Derived from Rudraksha Seeds (Elaeocarpus ganitrus) using U-Net

Bishwas Pokharel [a], Rajeshwar man Shrestha [b], Vasanta Gurung [c], Rinita Rajbhandari [d], Nanda Bikram Adhikari [e]

[a, e] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Nepal*
[b, c, d] *Department of Applied Sciences and Chemical Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Nepal*
[a] pokharel09.bsws@gmail.com, [c] vasantagurung@my.unt.edu, [e] adhikari@ioe.edu.np

Abstract
In computer vision and image processing applications, segmenting an image becomes an important task. The primary aspect of semantic segmentation is pixel-level labeling. In material science and Engineering, microscopy images show space information of matter, materials' morphology, phase, crystallography, magnetic structure, atomic structure, etc. Due to the complex pattern of microscopy images in this field, there is still a challenge to segment the various sections of interest. In this work, the U-Net architecture is used and this network is trained with high-resolution scanning electron microscope (SEM) images of activated carbon derived from Rudraksha seeds. The architecture consists of two parts: one that is contracting and another that is expansive along with skip connection. The Dice Coefficient is performed to evaluate the quality by matching the features with the ground truth images. The batch size of four and five is found to have lower test loss of 41% and 39%. Learning rates of 0.00001 are found to have the highest test dice acc of 61%, 62%, 64% at epoch 10, 50, and 150 respectively. Dropout of 0.2 rate slightly improves the performance metrics. Epoch 450, 800, 1000, and 2000 are found to have significant improvement over model performance. At epoch 800, the model achieves the best test dice score of 79 %. At epoch 2000 it achieves the best validation dice score 81%, the best test dice score is 79%, and the best test accuracy is 91%. The result reflects that even though the best accuracy is low as compared to the accuracy of the original paper (95%), the evaluated model can generate a segmented mask on the new unseen dataset.

Keywords
Semantic Segmentation, Microscopy Images, Activated Carbon, Rudraksha seeds, U-Net architecture, Dice Coefficient, Elaeocarpus ganitrus, Hyperparameters

1. Introduction

Image segmentation has become an important task in computer vision and image processing applications. Nowadays, it is widely used in a variety of fields, including medical image analysis, robotics, autonomous driving, augmented reality, video surveillance, and many more [1]. In the medical field with the advancement of technology, it is easy to capture medical images of different parts of the human body. The most commonly used high-tech equipment for capturing images are X-ray, ultrasound, Magnetic Resonance Imaging (MRI), and Computed Tomography (CT) but to help experts make better analyses and accurate diagnoses, segmentation of vital objects and extraction of essential features from captured images are necessary. A large number of papers have been published recording the success of fully automated segmentation of medical images based on deep learning [2].

Deep learning is now being used more widely in microscopy image processing for tasks like detecting nuclei, tissue segmentation, cell segmentation, image classification, and many others. Convolutional neural networks (CNNs) are the more common deep learning architecture used in computer vision and biological image processing tasks [3].

To look into the details of matter at the spatial scale of a micron and to take microscopic images, imaging equipment including optical microscopy, transmission electron microscopy (TEM), scanning electron microscopy (SEM), and scanning probe microscopy (SPM) is used. Real-time information about matter, including the morphology, phase with one another, crystal structure, magnetic structure, and molecular structure of materials, is provided by microscopy images. Material science also investigates the relationships among chemical or physical attributes [4]. In the field of material science, due to the rapid advancement in automation in experimental equipment and the tremendous collection of experimental and computational datasets; the size of publicly available datasets has increased significantly. The Materials Genome Initiative (MGI) and findable, Accessible, Interoperable, Reusable (FAIR) principles also contributed a lot. Because of this increasing volume of datasets, a fully automated analysis must be necessary; deep learning comes into play [5].

In this work, the datasets used are captured by Scanning Electron Microscopy(SEM). To study the nature of pores and their tentative concentration in SEM images of activated carbon prepared at different temperatures; deep learning is implemented.

2. Literature Review

The process of separating an image into multiple areas, with the desired region in one class and the other regions in a different one, is known as image segmentation. In essence, there are three methods for segmenting images: panoptic, instance, and semantic segmentation. Pixel-wise labeling is semantic segmentation, whereby each pixel is assigned to a specified label. Through instance segmentation, every pixel that is part of the regions of interest is distinguished. Semantic and instance segmentation combine to form panoptic segmentation. The objective is to give every pixel in an image a distinct instance ID and a semantic classification. Image semantic segmentation focuses on the pixel classification of an image. For such a goal, Encoder-decoder structures are widely adopted architecture, such as FCNs [6], U-Net [7], and Deeplab [8]. In these structures, an encoder brings out important image features, while a decoder brings back the extracted features localization and generates the segmented masks. The first high-impact encoder–decoder structure, Ronneberger et al. [7] U-Net has gained widespread acceptance among researchers for medical image segmentation. Long et al.[6] introduced Fully Convolutional Networks (FCNs), a benchmark for semantic segmentation tasks, using skip connections, which give a segmentation map in the output having the same dimension as input data.

Arbitrarily sized images are handled by modifying widely accepted CNN architectures, such as VGG16 and GoogLeNet. The authors evaluated their FCN model on three datasets: PASCAL VOC, NYUDv2, and SIFT Flow, and significantly improved segmentation performance and achieved impactful results on these datasets. The work has been widely cited and has made a significant impact on the field of computer vision. But this method too has some limitations: computationally expensive, inefficiently accounting for global context information, and is difficult to generalize to 3D images.

Motivated by [6], Ronneberger et al. [7] proposed U-net for microscopy image segmentation. It has two parts, a contracting path, and a symmetrically expanding path. It has features such as skip connections, a weighted loss for the separation of borders between touching objects, and elastic deformations for data augmentation. It is regarded as one of the best papers for segmentation in 2015 and was an advancement to learn successfully from a small number of annotated images (almost 30). It also won the ISBI Cell Tracking Challenge 2015 by a significant margin. U-Net has many modified variants and all of its variants are successfully implemented for a variety of images and problem areas. Zhou et al [9] proposed a nested U-Net, Cicek et al [10] introduced a U-Net design for 3D images and Zhang et al. [1] introduced a road segmentation method using pure U-Net architecture.

The Attention U-Net [11] was proposed by Oktay et al. in 2018 and includes a unique self-attention gating (AGs) filter and skip connections. This model's foundation is a VGG-16 with AGs/Resnet. This model's attention mechanism allows it to concentrate on specific areas of an image, making it very useful for jobs that need accuracy, like the multi-class abdominal CT-150 dataset.

3. Methodology

3.1 Dataset Collection

Rudraksha seeds were collected from the market of the Sankhuwasabha district. Activated carbon of Rudraksha seed was prepared in the laboratory at different temperatures. Scanning electron microscopy (SEM) known as The Hitachi S-4800 FE-SEM, in Japan, captures SEM images at different scales at a speeding voltage of 10 kV.

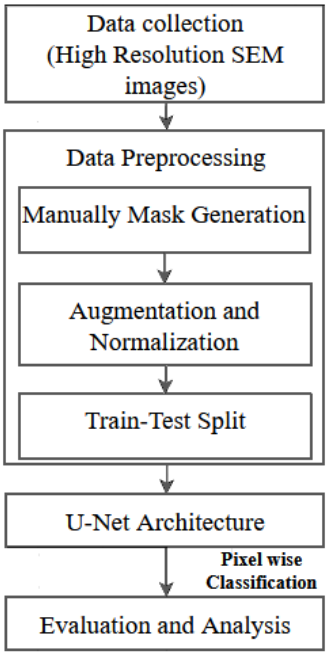

Figure 1: Workflow Diagram

.

3.2 Data Preprocessing

Corresponding masks of images were created manually by using the available open-source tools. The performance of semantic segmentation algorithms has been substantially enhanced by image augmentation. Image augmentation somehow prevents overfitting and enables the model to detect objects or regions of interest in a variety of instances by artificially increasing the training dataset using methods like rotation, scaling, flipping, and cropping. Normalization of grayscale images is done in order to adjust pixel values to a constant range, usually between 0 and 1. This ensures numerical stability, serves to speed up convergence, and enhances the model's capacity to extract useful features from the data during model training.

3.3 U-Net Architecture

The architecture shown in Fig. 2 consists of two parts: one that is contracting and another that is expansive. The contracting path is responsible for extracting important features like vertical edges, horizontal edges, etc. from the images. In the contracting path, several multiple operations such as convolution, ReLU, and max pooling are performed. Each layer of the contracting path in U-net operates on two 3×3 unpadded convolutions, both of which follow ReLU activation,

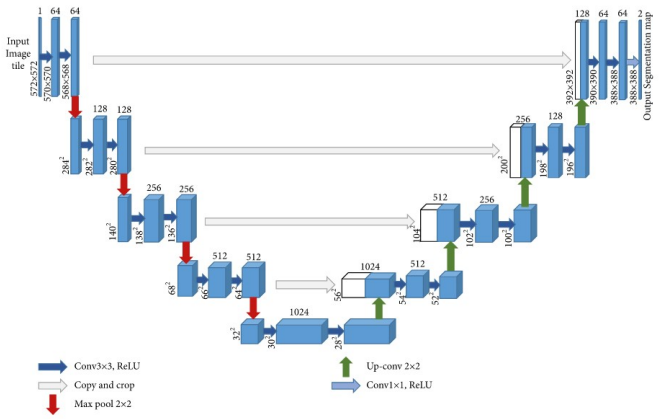

Figure 2: U-net architecture
(source: https://arxiv.org/abs/1505.04597)

and finally a 2×2 max pooling operation. The stride of 2 for max-pooling cut the dimension in half. Calculation of image dimension after each convolution in U-net:

$$n - f + 1 \tag{1}$$

n = image dimension, f = filter dimension.
In U-Net say, the input image has dimensions 572×572. After each convolution with filter 3×3, the output image has dimensions of $572 - 3 + 1 = 570$, $570 - 3 + 1 = 568$, and so on. The max pooling halved the dimension so dimension $\frac{568}{2} = 284$ is input for the next layer. After each max pooling, the size of the filter becomes double in its contracting path. 2×2 up-convolution results in a doubling of the image size and a halving of the channel size. For more accurate localization, skip connections enable the concatenation of feature maps from matching downsampling layers. After cropping, the contracting path's up-convolution and concatenation are used to calculate the dimension as follows: $(64 \times 64 \times 512)$ is cropped from the contracting path to $(56 \times 56 \times 512)$. As a result, $(56 \times 56 \times 512 + 56 \times 56 \times 512) = (56 \times 56 \times 1024)$ and can be seen in Fig.2.

Calculation of number of parameters in each layer of U-Net is expressed as:

- filter dimension, fd.
- depth (total image/s), d.
- no. of filters in the layer, nf.
- no. of bias, nb.
- Number of parameters, np.

$$np = fd \times nf \times d + nb \tag{2}$$

Calculation:

1. No. of depth =1, filter dimension = 3×3, 64 filters, and 1 bias = 64 bias for each filter = $(3 \times 3) \times 64 + 64 = 640$.
2. No. of depth = 64, filter dimension = 3×3, 64 filters, and 1 bias = 64 bias for each filter = $(3 \times 3) \times 64 \times 64 + 64 = 36,928$.
3. After max pooling first convolution, No. of depth = 64, filter dimension =3×3, 128 filters, and 1 bias = 128 bias for each filter = $(3 \times 3) \times 64 \times 128 + 128 = 73,856$ and so on.

Softmax and Cross entropy: The softmax function is used to calculate the energy function in combination with the loss function known as cross-entropy. It bounds the input to the range [0; 1].

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{3}$$

The loss function or cost function is used to evaluate how well a specific algorithm performs on the given training data. If the prediction or generalizing capacity of a model is poor, the loss is expected to be in large value. Training a model refers to optimizing the weights of learnable parameters weights by minimizing the loss. The loss compares the class predictions: a depth-wise pixel vector against a target vector for each particular pixel i.e. whether the pixel lies in the region of interest or background.

3.4 Metrics for Model

(a) Dice Coefficient: A well-liked statistic for measuring image segmentation algorithms is the Dice Similarity Coefficient (DSC), referred to as the Dice coefficient. It determines how much of the anticipated and actual binary maps overlap by comparing the overlap area to the total number of pixels in both maps.

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|} \tag{4}$$

$$\text{Dice} = \frac{2TP}{2TP + FP + FN} = F1 \tag{5}$$

When testing with binary maps with foreground as the positive class, the Dice coefficient and F1 score are comparable since they both achieve an appropriate balance between recall and accuracy.

(b) Intersection over Union (IoU): Semantic segmentation algorithms' efficacy is typically assessed using the Jaccard Index, also known as IoU (Intersection over Union). It calculates the percentage of the total area that both segments cover when the projected segmentation and the actual segmentation overlap. Using both accurate and incorrect predictions, IoU provides an assessment of how closely expected segmentation reflects the actual ground truth. The average IoU (mIoU) is the IoU across every class.

$$IoU = J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{6}$$

Here the ground truth map is B and the predicted segmentation map is A.

4. Results and Analysis

4.1 Data pre-processing and augmentation

To facilitate computations, the pixel values are normalized in the 0–1 range. After that, the datasets are center-cropped. The high-resolution images and masks are cropped to 512×512 pixels. Because this dimension is the normal input image size

for the architecture, the output must have the same dimension as the input, which is achieved by padding. Data augmentation is done using horizontal and vertical flips. Some collection of datasets after transformation is shown in fig 3.

According to the data in the table 1, the learning rate of 0.00001 seems to be the best among the possibilities. At all epochs (10, 50, and 150), this learning rate produced the greatest test dice accuracy. Additionally, the accuracy increased gradually over time, demonstrating the model's ongoing improvement. In contrast, the model performed extremely poorly when the learning rate was 0.1, with a test dice accuracy of 0.0 at both epochs 10 and 50. The model performed rather well with learning rates of 0.001, 0.000001, and 0.0000001, however, the gain in performance over epochs was not as steady as with the learning rate of 0.00001.

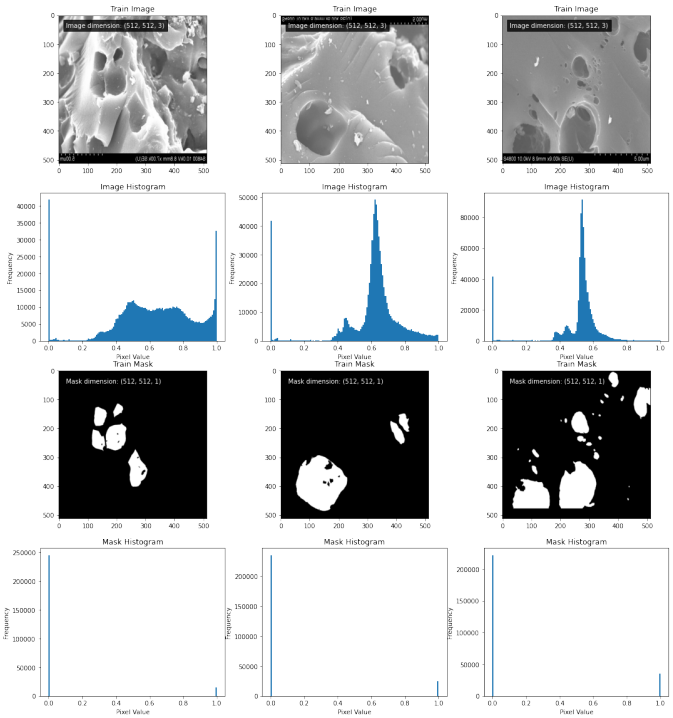

Figure 3: Augmented dataset (normalized and flip)

Table 1: test dice accuracy at different learning rates and epochs

learning rate	epoch	test dice acc.(×100) %
0.1	10	0.0
	50	0.0
	150	0.21
0.001	10	0.33
	50	0.53
	150	0.17
0.00001	10	0.61
	50	0.62
	150	0.64
0.000001	10	0.51
	50	0.52
	150	0.53
0.0000001	10	0.34
	50	0.47
	150	0.57

While a learning rate of 0.00001 resulted in the highest test accuracy, the high validation loss as shown in Fig.4 suggests that the model may be overfitting to the training data. To avoid overfitting, several techniques can be proposed, including early stopping, dropout, weight decay, and data augmentation [12]. The experiment already uses data augmentation. Figure 4 shows loss without any dropout and figure 5 shows a dropout rate of 0.2. However, it's important to note that the optimal hyperparameters for regularization techniques depend on the specific dataset and model architecture, and it may be necessary to experiment with different hyperparameters to find the best combination.

Figure 4: Epoch number vs. loss at 0.00001 learning rate without dropout

Figure 5: Epoch number vs. loss at 0.00001 learning rate with dropout

When a dropout rate of 0.2 is implemented then figure 5 shows it leads to a smaller gap between the training and validation loss, more effective at a learning rate of 0.00001. It suggests that dropout regularization has been effective in reducing overfitting. This can improve the model's ability to generalize to new, unseen data. However, using dropout regularization can potentially lead to a decrease in the quality of generated masks, as it may result in a loss of information and make it more challenging for the model to accurately generate masks [13].

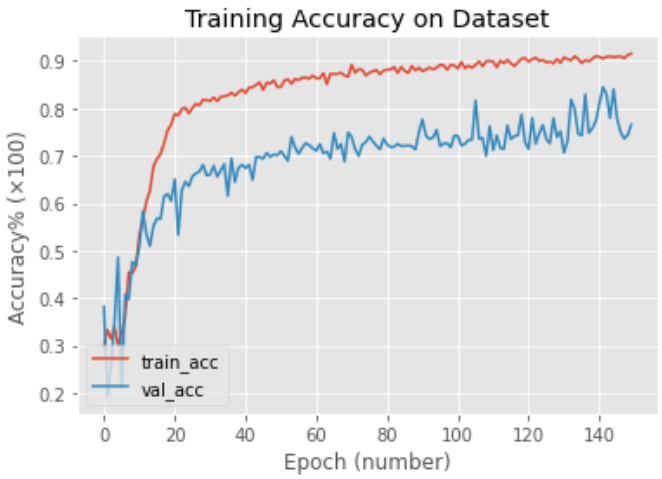

Figure 6: Epoch number vs. accuracy at 0.00001 learning rate

Table 2: Performance matrices at different epochs at a learning rate of 0.00001

Metric×100 (avg.%)	Epoch 50	100	250	350	450	550	800	1000	2000
min_train_loss	0.39	0.32	0.26	0.24	0.12	0.12	0.10	0.06	0.04
dropout(0.2)	0.57	0.41	0.30	0.24	0.34	0.30	0.20	0.17	0.14
min_val_loss	0.45	0.33	0.33	0.37	0.25	0.27	0.26	0.22	0.22
dropout(0.2)	0.62	0.43	0.37	0.30	0.36	0.33	0.29	0.28	0.24
max_train_acc	0.94	0.97	0.98	0.98	0.99	0.99	0.99	0.99	0.99
dropout(0.2)	0.80	0.91	0.92	0.95	0.92	0.98	0.94	0.95	0.97
max_val_acc	0.92	0.90	0.93	0.93	0.93	0.94	0.93	0.93	0.93
dropout(0.2)	0.75	0.87	0.89	0.92	0.89	0.87	0.89	0.88	0.87
max_train_dice	0.76	0.88	0.94	0.95	0.97	0.97	0.98	0.98	0.98
dropout(0.2)	0.32	0.52	0.59	0.77	0.60	0.89	0.74	0.76	0.81
max_val_dice	0.63	0.66	0.76	0.74	0.72	0.78	0.74	0.75	0.81
dropout(0.2)	0.41	0.52	0.51	0.64	0.52	0.43	0.55	0.61	0.65
test_loss	0.69	0.43	0.43	0.42	0.45	0.47	0.57	0.32	0.29
dropout(0.2)	0.60	0.42	0.36	0.41	0.36	0.34	0.28	0.25	0.26
test_acc	0.83	0.88	0.89	0.90	0.87	0.87	0.85	0.90	0.91
dropout(0.2)	0.81	0.89	0.86	0.86	0.89	0.88	0.89	0.91	0.90
test_dice	0.55	0.63	0.68	0.71	0.58	0.64	0.79	0.79	0.79
dropout(0.2)	0.59	0.50	0.72	0.63	0.65	0.59	0.75	0.80	0.81

For a better understanding of the nature of graph and model performance, different metrics are taken for performance as shown in Table 2. The minimum training loss decreased over the course of the epochs, indicating that the model was getting better at fitting the training data. The minimum validation loss also decreased over time, although not as consistently as the training loss. This suggests that the model was improving on unseen data as well, but may have experienced some fluctuations due to random variations in the validation set or the training process. The maximum training accuracy and dice score (another metric commonly used in segmentation tasks) increased over time, which is expected since these metrics measure how well the model fits the training data. The maximum validation accuracy and dice score also increased over time but with more fluctuations than the training metrics. This suggests that the model was able to generalize to unseen data to some extent, but may have hit some plateaus or faced some challenges in improving further. Finally, the test metrics (loss, accuracy, and dice score) were evaluated on a test set and generally showed good performance. The best hyperparameters (i.e., the ones that achieved the highest validation dice score) were able to achieve a test dice score of 0.79, which is a decent performance.

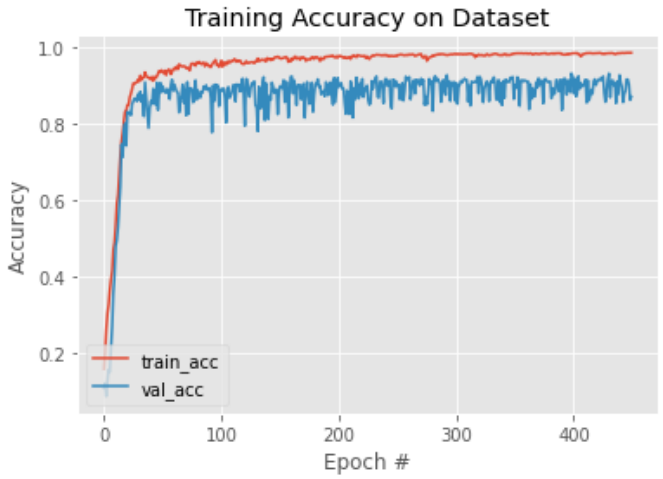

Figure 7: Epoch number vs accuracy at learning rate 0.00001

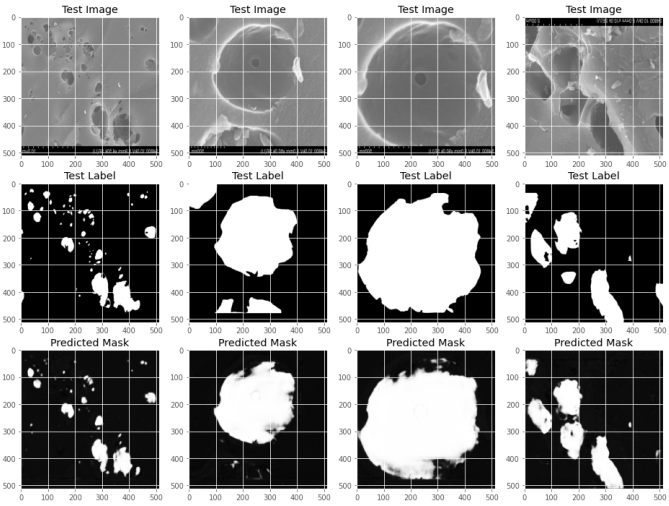

Figure 8: Predicted mask at Epoch 450 at learning rate 0.00001

It can be seen that the training accuracy is always greater than the validation accuracy and the nature of training and validation loss are similar for different batches respectively. In the Overall scenario, metrics values do not improve drastically and do not get stabilized on increasing the batch sizes besides the training speed.

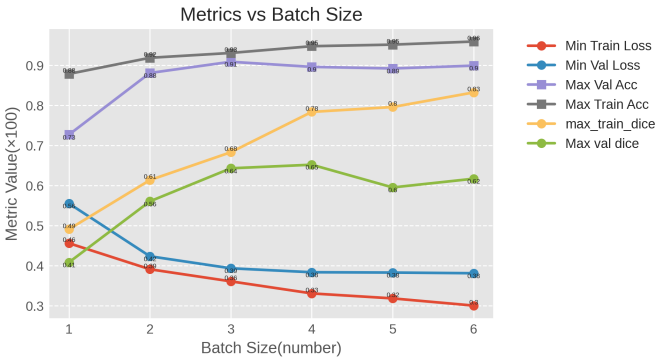

Figure 9: Different metric values vs batch size for training and validation dataset

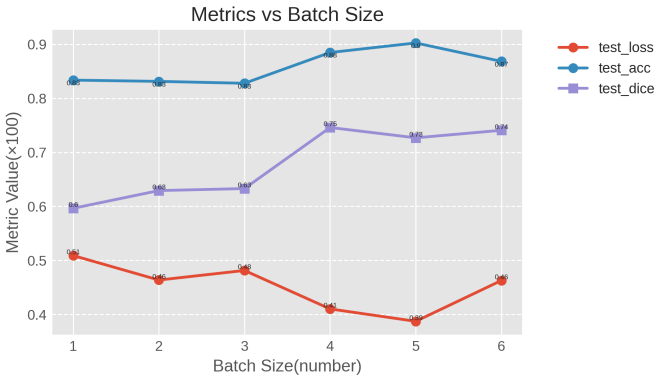

Figure 10: Different metric values vs batch size for test dataset

5. Conclusion

The U-Net architecture consisting of the contracting and expansive parts is trained with high-resolution SEM images of activated carbon derived from Rudraksha seeds. The data augmentation and normalization are done. The tuning of different hyperparameters such as batch size, learning rate, epoch number, and dropout rate are performed. The batch size of four gives slightly better performance metrics (not significance difference) so it is considered for further analysis. Among various learning rates, 0.00001 is found to have high test dice accuracy without a drop rate and with a drop rate of 0.2. Epoch 450 shows a significant improvement in the training loss compared to earlier epochs, indicating that the model is fitting the training data better. However, the validation loss and accuracy do not show much improvement, suggesting that the model may be overfitting to the training data. Furthermore, the model achieves the train dice of 0.97 % best test dice score (0.79) at this epoch. The result reflects that even though the accuracy is low as compared to the accuracy of the original paper [7] which is almost 95%, the evaluated model can produce a segmented mask.

Acknowledgments

The authors are grateful to the Department of Electronics and Computer Engineering of the Pulchowk Campus and colleagues Anjuli Sapkota, Bhimraj Yadav, and Deep Shankar Pandey for their support and cooperation.

References

[1] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[2] Tao Lei, Risheng Wang, Yong Wan, Xiaogang Du, Hongying Meng, and Asoke K Nandi. Medical image segmentation using deep learning: A survey. 2020.

[3] Fuyong Xing, Yuanpu Xie, Hai Su, Fujun Liu, and Lin Yang. Deep learning in microscopy image analysis: A survey. *IEEE transactions on neural networks and learning systems*, 29(10):4550–4568, 2017.

[4] Mengshu Ge, Fei Su, Zhicheng Zhao, and Dong Su. Deep learning analysis on microscopic imaging in materials science. *Materials Today Nano*, 11:100087, 2020.

[5] Kamal Choudhary, Brian DeCost, Chi Chen, Anubhav Jain, Francesca Tavazza, Ryan Cohn, Cheol Woo Park, Alok Choudhary, Ankit Agrawal, Simon JL Billinge, et al. Recent advances and applications of deep learning methods in materials science. *npj Computational Materials*, 8(1):1–26, 2022.

[6] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[9] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018.

[10] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, et al. U-net: deep learning for cell counting, detection, and morphometry. *Nature methods*, 16(1):67–70, 2019.

[11] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[12] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.

[13] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.