Technical University of Denmark

Summer Student Research Project

U-Net Artificial Intelligence for the Automated Segmentation of Solid Oxide Cell Anode Microscope Images

Richard S. Chiu

July 3, 2023

Project Advisors:
Dr. Peter Stanley Jøgenson
Dr. Salvatore De Angelis
Department of Energy Conversion and Storage

1 Abstract

We live in a world of advanced technologies and rapid development, but as we strive for innovation, scarcity of resources becomes just as much of a consideration. Lithium-ion batteries have a significant foothold in the energy marketplace, with its size being US $45.7 billion in 2022. It is estimated to have a compound annual growth rate of 13.1%, reaching US $135 billion by 2032. However, building these lithium-ion batteries are costly as they use materials in short supply, such as lithium, nickel, cobalt, and manganese.

A possible alternative to Lithium-ion batteries for energy storage could be Solid Oxide Cells (SOCs). They are solid ceramic devices that convert chemical to electrical energy. SOCs are efficient, combustion-less, virtually pollution-free power sources that run almost silently with few moving parts. However, a breakthrough in the energy market for SOCs is stunted by degradation within the cell. Both performance and durability heavily depend on the microstructure design and material distribution throughout the cell. So far, some of the analysis of this process involves the manual procedure of a trained individual annotating thousands of microscopy images. This arduous process requires a significant amount of time and effort, manpower that could instead directed into producing higher-performing and durable SOCs.

The goal of my project is to simplify the automated image segmentation that identifies different phases that exist throughout the SOC's microstructure. The power of artificial intelligence (AI) is leveraged in this code to solve the complex segmentation process. Both of my codes use a pre-trained fine-tuned U-Net model that has achieved an accuracy of Intersection over Union (IoU) of 97.9%, indicating the promise of using AI for automated imaging segmentation.

Table of Contents

2 Introduction

The aspiration for green power generation is always gaining more attention, making the fuel cell family garner even more interest and attention. The solid oxide cell (SOC) demonstrated attractive characteristics such as high efficiency and lower emissions of sulfur and nitrogen oxides, hydrocarbon pollutants, and significantly lower CO_2 emissions. SOCs could be an alternative to batteries for large-scale energy storage [1] but have shown some stability and degradation problems. The 3D electrode microstructures of these SOCs need to be properly designed as both reliable and durable [2,3]. This central issue needs to be solved before being commercially widespread and usable to produce clean energy.

2.1 Background of Solid Oxide Cells

Solid oxide cells (SOCs) use a solid ceramic oxide for the electrolyte. By reacting with fuel gases such as hydrogen, methane, and carbon dioxide in an electrolyte, SOCs can produce electricity by oxidizing fuel, and store electricity by creating fuel.

All components in SOCs are solid. With fewer moving parts, noise and maintenance are reduced. SOCs are highly efficient, and able to operate on a very wide variety of fuel types. SOCs are also resilient to high temperatures, being able to perform normally and even use excess heat to their advantage in hybrid systems.

2.2 Project Scope

My goal for this project is to modify and achieve pre-trained model code performance comparable to code trained from scratch in segmenting microscopy images. The goal is to achieve a mean intersection over union (IoU) of 90% or above. Using a pre-trained model with good certainty helps simplify the process of segmenting SOC images rather than manually training the code each time. All my code used in this project can be found here [15].

2.2.1 Semantic Segmentation

Semantic segmentation is a computer vision task. An example is provided in Figure 1. Each pixel of an image is labeled and identified with a class of what is being represented. In other words, the computer should be able to scan and mark distinct features of a given image. For my project, the images would be the SOC microstructures.

Figure 1: Semantic segmentation of an image to identify the people, cars, sidewalks, and regulatory signs. Figure from [11].

As demonstrated in Figure 1, semantic segmentation algorithms analyze digital images and identify different objects by labeling individual pixels belonging to that category. This example shows a busy street with many different objects recognized and segmented such as the people, cars, sidewalks, and regulatory signs.

2.3 Project Limitations

The Deep Neural Networks used have already been trained with the U-Net model. Further training was required since it is unknown what these Neural Networks (NNs) were previously trained to recognize. NNs and their performance are still highly dependent on the data. The accuracy of the segmentation will still heavily depend on the data sets given to the NNs.

Only 128x128 pixel images are supported in the system. Using a Carl Zeiss Supra Scanning Electron Microscope with a secondary electron detector, polished, 2D, cross-sectional images can be obtained using the scanning electron microscopy (SEM) technique. Clear distinction and good samples of the SOCs microstructure will greatly impact the training and following of the Deep Neural Networks.

This system has also been restricted to the segmentation of the metallic nickel phase. These regions are easily identified by a human as the layer contains grains that appear to be raised at a

higher level than the rest of the medium. This program is therefore limited to binary segmentation, only able to identify two features rather than multiple features.

3 State of the Art in Image Segmentation

Image segmentation can be traced back to the past three decades, since then rapidly evolved in application across many fields [4]. The following chapter will cover the techniques and technologies used in this code to solve the complex problem of segmenting SOC microstructure images, subdividing an image into its parts, and labeling those parts as groups of pixels.

3.1 Deep Learning Algorithms

Stemming from Machine Learning, Deep Learning makes use of neural networks having three or more layers. Deep Learning mimics the behavior of the human brain to adapt and learn from large data sets through a combination of data inputs, weights, and biases. These processes work together to accurately recognize, sort, and describe objects in the data sets.

Deep neural networks consist of many layers that are interconnected through nodes. Each node will build upon another, refining and optimizing the previous layers. This process of computation is called forward propagation. Both the input and output layers within the neural networks are called visible layers. The input layer is where the data is introduced, and the output layer is where the predictions are made, and classification given. Processes like backpropagation also use algorithms to calculate errors in predictions by adjusting weights and biases. This is achieved by the algorithm referencing previous layers that minimize errors from the prediction. By utilizing both forward propagation and backpropagation, the algorithm can make the best predictions by correcting for any errors while processing. Given time, these algorithms can learn extremely quickly while minimizing their errors.

While the above describes some of the simplest types of deep neural networks, these networks can become incredibly complex. Different types of neural networks are tailored to tackle very specific problems, with some only working with certain data archetypes. Below are some examples,

3.1.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are primarily used for computer vision and image classification. These applications detect features and patterns from an image, allowing for tasks

like object detection and recognition. CNNs are artificial neural networks based on Deep learning algorithms for their image classifications and segmentations.

This algorithm is part of the supervised deep learning methods. Therefore, training data sets are required. With those data sets the convolutional neural network models can predict binary segmentation masks from any image not from the training set.

CNNs and their architecture can be organized into different layers:

- Convolutional Layer: The main layers in a CNN where the most computation takes place; these layers are in charge of extracting key image features. The CNN is made up of a series of elements: input data, a series of filters or kernels, and feature maps.
- Pooling Layer: Pooling layers oversee reducing the spatial resolution of each input. This is required to decrease the computation power needed to process the data. The pooling layer serves to progressively reduce the spatial size of each representation, reducing the number of parameters and amount of computation in the network, hence controlling overfitting. The grouping uses a kernel that operates in a similar way as the kernels in the convolutional layers. Normally the algorithm would use these two types of pooling operations:
    - Average Pooling: Average values of the receptive section of the feature map is collected and then stored into the output matrix.
    - Max Pooling: Maximum values in the receptive section are collected from the input and then stored in the output matrix.
- Fully Connected Layer: is the final layer and is also known as the hidden layer of the convolutional neural network. The goal of this layer is to classify the learned features from each image that the previous layers (convolution and pooling) have learned. This layer comprises an affine function and a non-linear function to calculate the class distribution for each pixel analyzed.

3.1.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are utilized for tasks relating to natural language and speech recognition, allowing specialization in processing sequential and time series data sets.

3.2 U-Net Architecture

U-Net is specifically designed for semantic segmentation. It associates a label/category with every pixel in an image. The architecture has two paths: a contracting path and an expansive path. The contracting path shares similarities in architecture to a convolutional network. It consists of repeated applications of two 3x3 convolutions (unpadded convolutions), followed by a rectified linear unit (ReLU), and a 2x2 max pooling operation with stride 2 for down-sampling. Next is the expansive path which up-samples the featured map. Each is followed by a 2x2 convolution ("up-convolution") that halves the number of feature channels. This process causes images to be cropped but is necessary due to the loss of border pixels in every convolution. The final layer displays the results in a 1x1 convolution to map the 64-component feature vector to the desired number of classes. In total, this network consists of 23 convolution layers.

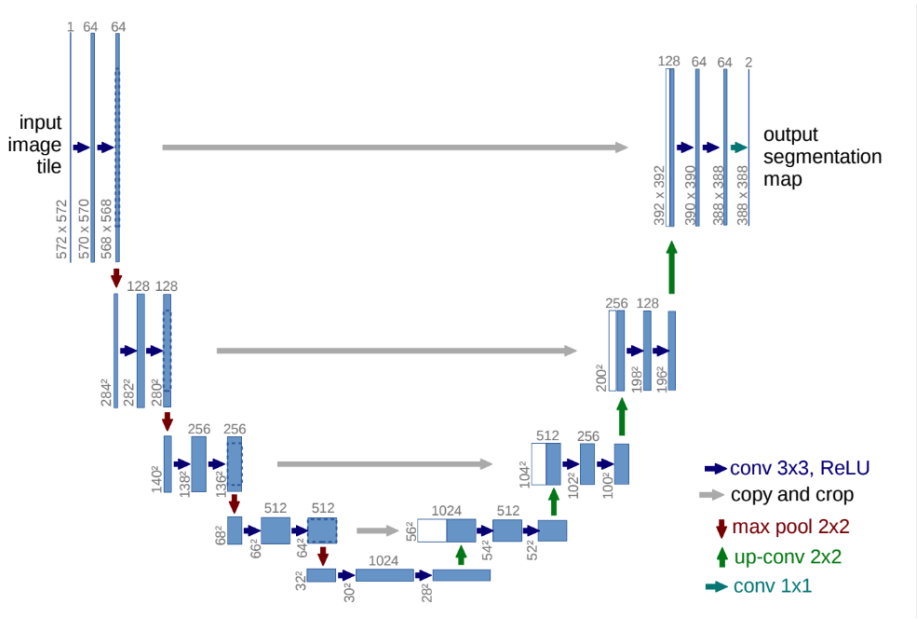

Figure 2: U-Net Architecture. Figure from [12].

Figure 2 shows a U-Net architecture that is based on fully convolutional networks. Its learning process, resulting from its modified architecture, extends U-Net so that can work with very few training images. This, in turn, can yield more precise segmentations. [5]

3.3 Pre-trained U-Net Models

Pre-trained models have the advantage of already being prepared and equipped to handle tasks. In this case, the training will benefit this problem to segment SOC microscopic images. Already

being pre-trained means the model inherently carries resultant weights and biases that may affect the desired results. To use the pre-trained model, we must further train the model and help specialize it to our domain of application.

Some of the most well-known and utilized pre-trained models for image recognition and segmentations are AlexNet, VGG, GoogLeNet, ResNetXt, SqueezeNEt, ResNet, Densenet, SuffelNet V2, and MobileNet V2.

My project will focus on only two pre-trained models: ResNet50 and VGG16. This work builds upon Sergio Segura's DTU Bachelor project [6].

3.3.1 Pre-trained ResNet50

The name ResNet stands for Residual Network and is a very specific type of CNN. ResNet-50 is a CNN with 50 layers of CNN (48 convolution layers, one MaxPool Layer, and one average pool layer). The architecture almost resembles the stacking of blocks, as residual neural networks are a type of artificial neural networks that form networks by stacking residual blocks.

ResNet-50 provides a way to add more convolution layers to a CNN without running into the vanishing gradient problem. By using the concept of shortcut connections, the connections skip over layers, converting a regular network to a residual network.

The regular network was based on VGG neural networks (ex: VGG-16 and VGG-19). However, ResNet has fewer filters and is less complex than a VGG network.

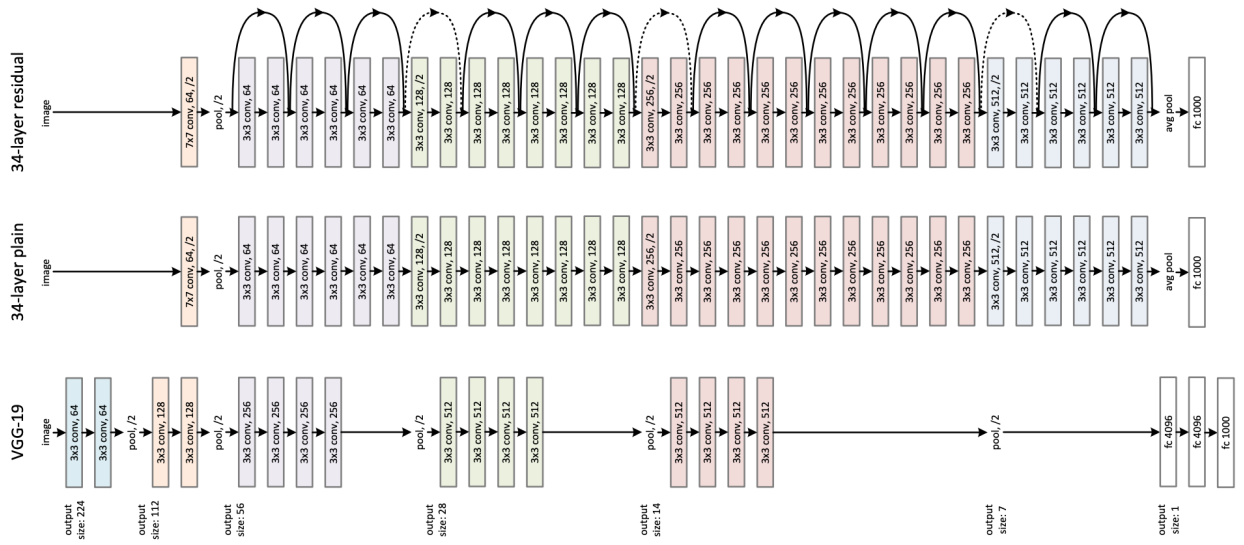

Figure 3: ResNet50 Architecture. Figure from [10].

The ResNet architecture adheres to two rules. First, the number of filters in each layer must stay relative to the size of the outputting feature map. Second, if the feature map's size is halved, the number of filters doubles to maintain the time complexity of each layer. At heart, the ResNet-50 architecture (shown in Figure 3) is similar to the architecture of VGG-16 (shown in Figure 4). ResNet has been proven to work well in segmentation tasks specifically as a pre-trained model utilizing a U-Net architecture [7].

3.3.2 Pre-trained VGG16

Introduced at the ILSVRC 2014 Conference, VGG-16 is one of the most popular pre-trained models for image classification. The Visual Graphics Group at the University of Exford developed VGG-16 as a successor to AlexNet, quickly being adopted by researchers and the industry for image classification tasks [8].

3.3.3 Code Repository

All code written for this project was developed in Python through Microsoft's Visual Studio code editor. All the codes and this project report are available on GitHub [15].

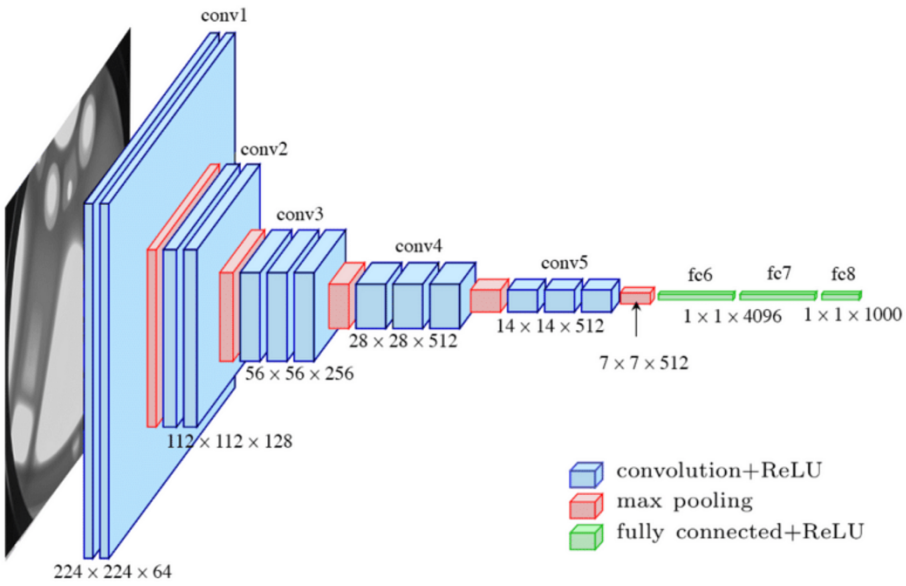

Figure 4: VGG16 Architecture. Figure from [13].

The model is sequential, using many filters like checks and balances throughout its stages. Each starts at a small 3x3 filter used to reduce the number of parameters as all the hidden layers use the ReLU activation function. Even after the filter, the number of parameters is 138 billion, making it a much slower model to train compared to others. [9]

Although specifically designed for classification tasks, it has demonstrated excellent performance as a pre-trained model with U-Net architecture. [10] This is for similar segmentation tasks.

4 Results and Discussion

Below will be the results from the two different algorithms developed to automate the segmentation of SOC microscopy images. Deep learning algorithms, their architecture, training, and segmentation predictions of both models will be included. Metrics such as IoU are also listed to measure model performance. Here, we can finally show how the U-Net model both works and performs at deeper levels through visualizations such as K-means Clustering.

4.1 Deep Learning Models Architecture

Presented is the implemented architecture of two different Deep Learning models. Both ResNet-50 and VGG-16 share very similar architecture, but results are still dependent on the methods of grid search performed for optimization.

4.1.1 Pre-trained U-Net: ResNet50

- Encoder: ResNet50 is the base model, consisting of 4 convolutional blocks, each having 3 convolution layers with 3x3 kernels followed by a ReLU activation and batch normalization layers. The main difference compared to other models, the activation and batch normalization layers are not applied directly, rather, both are applied after the addition of ResNet's residual blocks. After each block, the max pooling operation is the computer for down-sampling. Kernel increments increase from 64 to 1024.
- Bridge: Apex or deepest part of the model, the bridge has a convolutional block that connects both the encode and decoder. 2048 kernels are used in the block.
- Decoder: The decoder mirrors the encoder. The 4 decoder blocks start with a transpose convolution layer used for up-sampling. The result of this operation produced feature maps from the encoder, creating the skip connection. There are 3 convolution layers after, each of them followed by a ReLU, batch normalization, and a dropout layer. Kernel count halves with each level of depth, from 512 to 32.
- Final Layer: One convolutional layer with a single filter; it computes a binary prediction using a sigmoid function.

A Total of 71 million parameters are used. 23 million are trainable, and 48 million remain frozen while training to use the weights and biases of the ResNet50 as a base for the encoder.

4.1.2 Pre-trained U-Net: VGG16

- Encoder: VGG16 is the base model, consisting of 4 convolutional blocks:
  - The first two blocks each have two convolutional layers with a 3x3 kernel, followed by a ReLU activation layer and dropout layer.
  - The rest of the blocks follow this structure: 3 convolution layers, followed by a ReLU activation layer and a dropout layer.

Following each block is down-sampling with a max pooling of a 2x2 size with a stride of 2. Kernel increments double their values from 64 to 512 per level of depth.

- Bridge: Apex or deepest part of the model, the bridge has a convolutional block that connects both the encode and decoder. 512 kernels are used in the block.
- Decoder: The decoder makes use of 4 blocks, all of them starting with a transpose convolution layer used for up-sampling, followed by a batch normalization and dropout layer at a 0.4 rate. These operations are simultaneously performed with the feature maps from the encoder, creating the skip connection. Once again, the 2 convolutional layers are then followed by a ReLU, batch normalization, and dropout layer. The Kernel count halves with each level of depth going from 256 to 32.
- Final Layer: A 2D convolutional layer computes the segmentation map using the sigmoid activation function.

In total, this pre-trained model has 25 million parameters, with approximately 15 million frozen (inherited from the VGG15 model), leaving 10 million parameters that are trainable.

4.2 Deep Learning Model Training

This section presents the training graphs for both ResNet-50 and VGG-16 U-Net models I have trained in the project. Both models have already achieved low validation losses with high accuracies by the first 20 epochs as shown by the next two figures below. This trend appears for the ResNet-50 model but is even more exaggerated with the VGG16 model. In these sections, we will analyze these graphs, compare both models, and explore why there are struggles in generalization.

4.2.1 Pre-trained ResNet50:

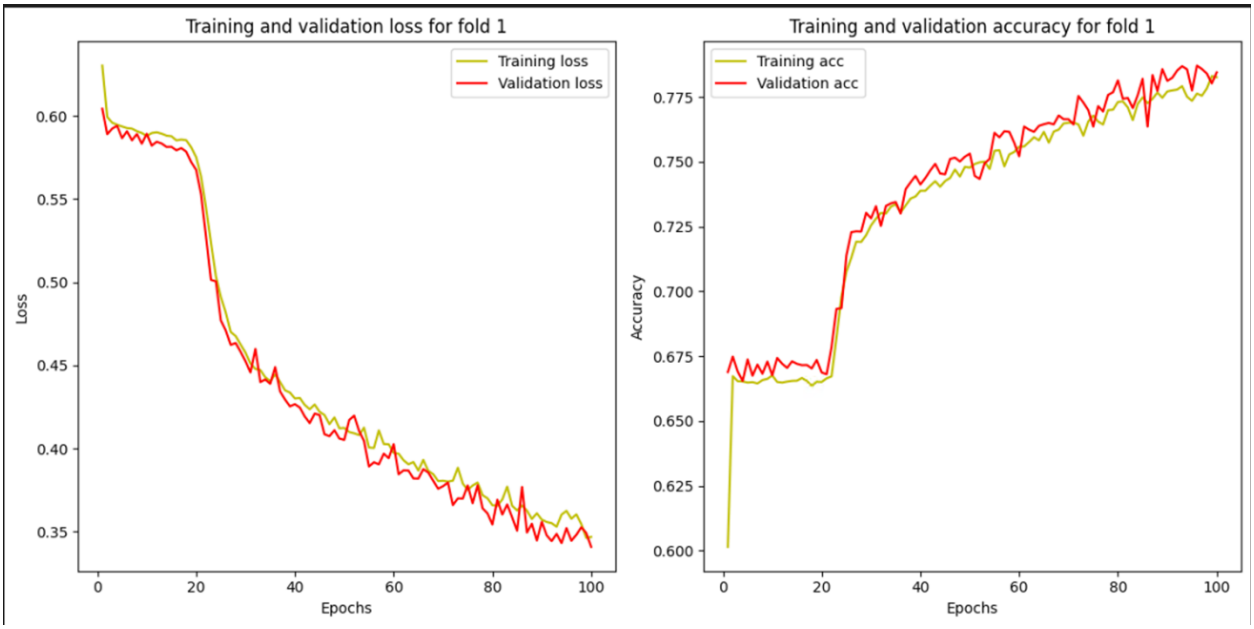

Figure 5: ResNet50 Training Curve (Loss and Accuracy).

In Figure 5, the training loss decreases while training accuracy increases as the Epochs progress. This is expected from a model that is correctly learning how to segment images given in the training dataset. Minor spikes in the validation loss curve appear, indicating the model is learning to generalize correctly. The symmetry of the red spikes to the accuracy graph reinforces the model's adaptability. Notice by epoch 50, a drop in validation occurs which indicates there may be a slight overfit in the model.

4.2.2 Pre-trained VGG16:

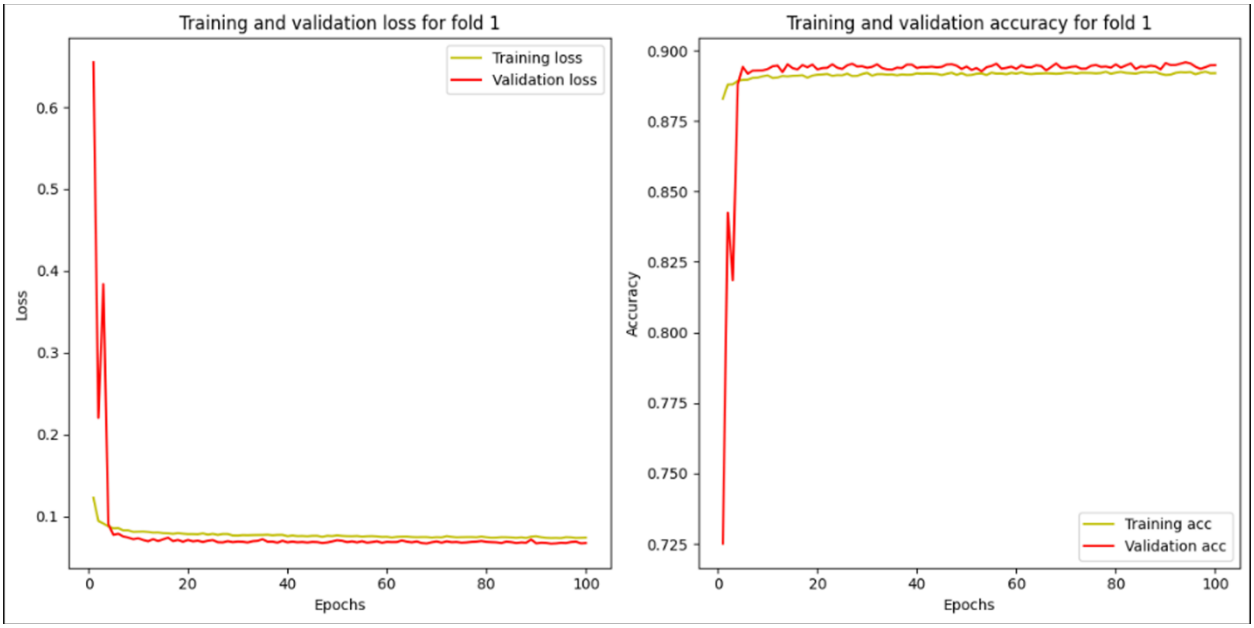

Figure 6: VGG16 Training Curve (Loss and Accuracy).

Similar to ResNet-50, the VGG-16 model seems to learn properly too, training loss decreases while training accuracy increases as shown in Figure 6. A small spike of validation loss occurs by around epoch 5, by normalizes slowly in the following epochs. Although there seems to be a permanent validation loss past epoch 5, we could argue the model is suffering from overfitting as a small gap between both graphs is still present, indicating a very minor struggle in generalization.

4.3 Deep Learning Models Segmentation Results

Before diving into the segmentation results, we need to explain the Intersection Over Union metric, as it plays a very important role in determining model performance.

IoU, also known as the Jaccard Index, is a metric that measures the percent of overlap between the prediction model and the ground truth given by manually annotated masks as shown in Figure 7.

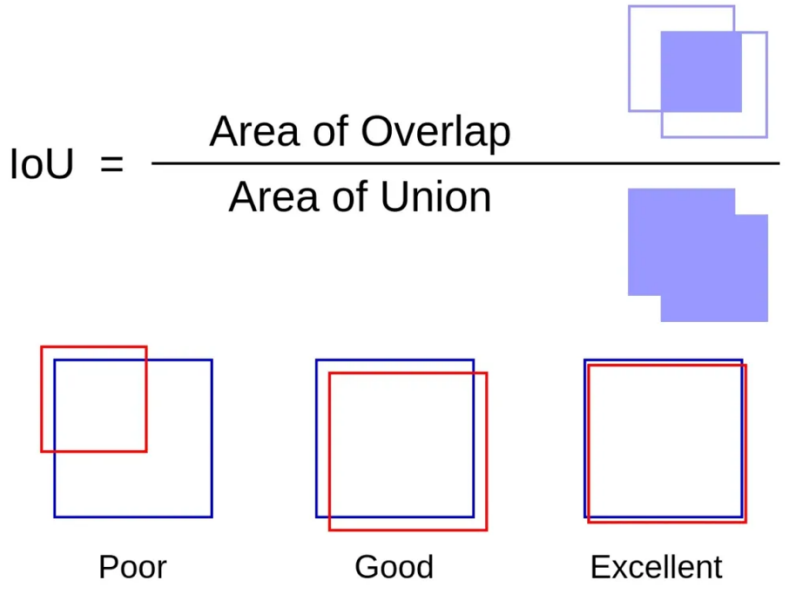

Figure 7: Calculating the Intersection Over Union (IoU) metric. Figure from [14].

In context to my project and binary semantic segmentation, IoU will measure how close the nickel phase segmentation predictions are to the ground truth. The closer to 1, the better the segmentation and performance the model has provided.

4.3.1 Pre-trained ResNet50:

Using a given SEM image (Testing Image), a ground truth (Testing Label) is manually created for every slice. This ground truth is compared to an equivalent segmentation prediction generated by ResNet50 (Prediction on Test Image). These results are shown in Figure 8 as the Testing Image, Testing Label, and Prediction on Test Image, respectively.

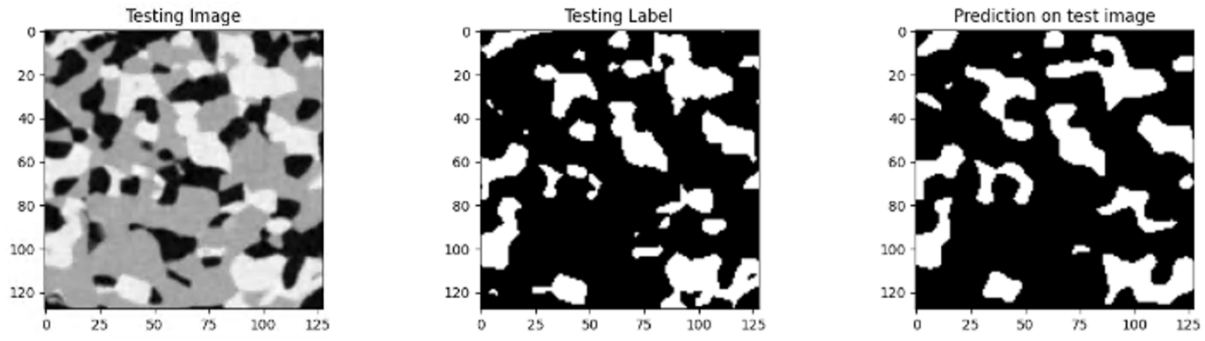

Figure 8: An experimental testing image obtained by SEM, the ground truth, and prediction by ResNet50.

The average IoU of the 250 images is 0.84. The model is sufficient in segmentation, and able to match most of the nickel phase structure. The majority of error comes from including and excluding grains not found around the original nickel phase layer. Note that a 0.84 IoU is fair performance, as our target accuracy is 90% or above.

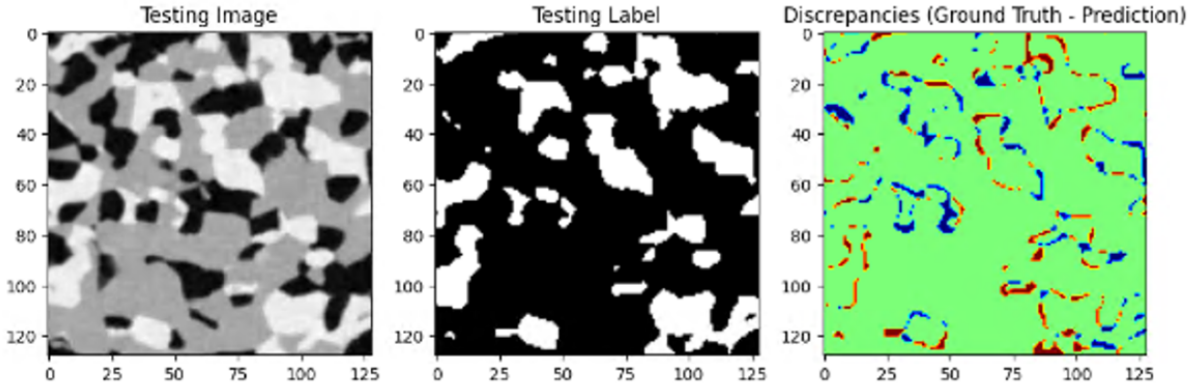

Figure 9: A comparison of an experimental testing image obtained by SEM, the ground truth, and discrepancies predicted by ResNet50.

The discrepancy plot in Figure 9 clearly shows how the model performs. Both the ground truth and ResNet-50's predictions are plotted, demonstrating how this model struggles to properly segment along the edges of the lighter nickel layers.

4.3.2 Pre-trained VGG16:

The given SEM image (Testing Image) and the ground truth (Testing Label) used in this study is shown in Figure 10. The ground truth is compared to an equivalent segmentation prediction generated by VGG-16 (Prediction on Test Image) shown in the right image of Figure 10.

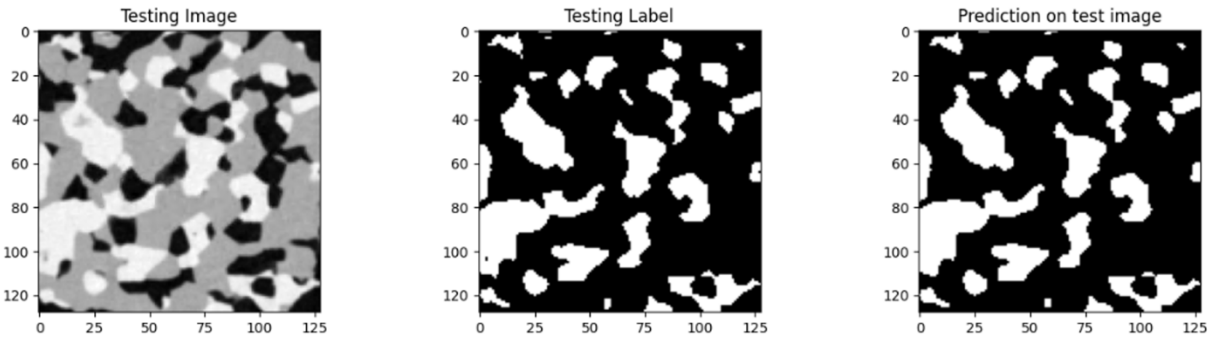

Figure 10: An experimental testing image obtained by SEM, the ground truth, and prediction by VGG16.

The VGG-16 model shows improved capability to segment images of SOC microstructures compared to ResNet-50. To the human eye, the segmentations shown in Figure 11 are almost perfect. There are some extremely minor variations around the pores and edges connecting the grains, but VGG-16 has achieved an almost perfect accuracy more clearly shown by the discrepancies between the ground truth (Testing Label) and model predictions (Prediction on Test Image).

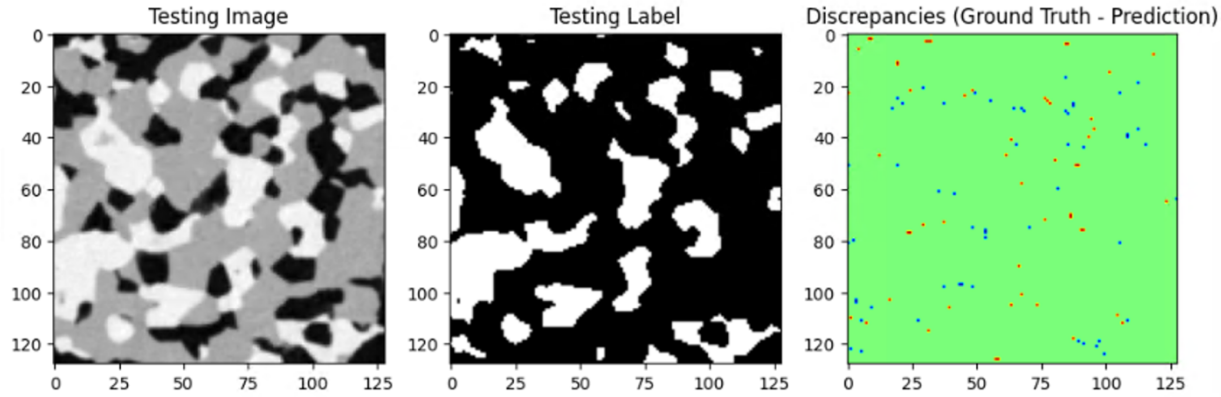

Figure 11: A comparison of an experimental testing image obtained by SEM, the ground truth, and discrepancies predicted by VGG16.

The average IoU of the 250 test images is 0.98. VGG-16 has performed an outstanding job, with almost perfect prediction as reflected in both its 0.98 IoU, with an almost flawless discrepancy plot.

4.4 Overall Comparison

From all the results, the U-Net VGG16 model (mean IoU of 0.979) outperformed the ResNet50 model (mean IoU of 0.861). Both produced good results given previous pre-trained models have yielded IoU values of 0.780. The total accuracy of my models equals 0.910, just above the target IoU of 90%. Results indicate that the pre-trained U-Net models exhibit a good balance between precision and recall.

5 Conclusions and Future Work

My project scope was to find a way to replicate and automate the process of segmenting the nickel phase of 128x128 SOC microstructure images. During my research, I was able to successfully study and implement different machine learning and deep learning algorithms for

image segmentation using the U-Net architecture. The U-Net model, VGG-16, trained with 250 images, performed incredibly well. I was able to automate the process of segmentation with outstanding results reflected by a 97.9% average intersection over the union (IoU), an accuracy well above the targeted 90% IoU. The pre-trained VGG-16 model outperformed every other model either implemented or researched throughout my project. During experimentation, I realized some issues with implementing the model, such as the limitations of transfer learning and the importance of skip connections in this type of image segmentation. Finally, I was able to extract and export trained VGG-16 and ResNet-50 models as large hdf5 files for other researchers or departments at DTU to use as a blueprint to segment images of their own.

5.1 Future Work

Excellent results were achieved, and this allows us to open a few possibilities based on the models implemented and trained to investigate and improve the current best model.

- Larger, different, and longer datasets: This study only investigated 128x128 pixel images. Would it perform better with larger data sets? Could this model work with different-sized photo resolutions or even different-sized data sets that are not monochrome and dualistic, but colored? A possible way of extending this project could be exploring these questions and training the model with a broader dataset that will deal with various types of images.
- Imprecisions at the edges of images and image size: Though testing we know that models like U-Net perform worse near image edges. This is due to the convolutional and pooling operations not giving the filters enough information at the edges, causing a loss in spatial information. Techniques such as mirrored padding and specialized loss function deal with and explore these kinds of problems more in-depth.
- Other DNN models: This study only focused on VGG-16 and ResNet50. It will be worthwhile to try other models utilizing the same U-Net architecture for semantic segmentation of SOC images, e.g. GoogLeNet, ResNetXt, and SqueezeNEt.
- Multiclass models: Due to the time constraints of this project I only considered binary segmentation. Colored image segmentation exists and can be explored in the future, annotating images and training multi-class models using U-Net architecture for segmentation is possible.
- Further improvements: Even more possibilities of investigation can include adjustments in the model architecture, its depth, and even testing other pre-trained models.

6 References Cited

1. Emergen Research, "Lithium ion Battery Market By Material (Cathode Material, Anode Material, Electrolyte Material, Others), By Product Type (Component and Portability), By Voltage, By Vertical, and By Region Forecast to 2032," Report ID ER_001780, March 2023, https://www.emergenresearch.com/industry-report/lithium-ion-battery-market#:~:text=The%20global%20lithium%2Dion%20battery,13.1%25%20during%20the%20forecast%20period.
2. Department of Energy Office of Fossil Energy and Carbon Management, "Solid Oxide Fuel Cells," https://www.energy.gov/fecm/solid-oxide-fuel-cells.
3. R. M. Ormerod, "Solid Oxide Fuel Cells," *Chem. Soc. Rev.*, vol. 32, pp. 17-28, 2003, DOI: 10.1039/B105764M.
4. Y. Guo, Y. Liu, T. Georgiou, M. S. Lew, "A Review of Semantic Segmentation using Deep Neural Networks," *Int. J. Multimed. Inf. Retr.*, vol. 7, pp. 87-93, 2018, DOI: 10.1007/s13735-017-0141-z.
5. O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," arXiv:1505.04597, 2015, DOI: 10.48550/arXiv.1505.04597.
6. S. Segura, "Image Segmentation of Solid Oxide Cells Microstructures," Bachelor Project, Technical University of Denmark (DTU), June 2023.
7. S. Alam, N. K. Tomar, A. Thakur, D. Jha, A. Rauniyar, "Automatic Polyp Segmentation using U-Net-ResNet50," arXiv:2012.15247, DOI: 10.48550/arXiv.2012.15247.
8. K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556, DOI: 10.48550/arXiv.1409.1556.
9. D. Lin, Y. Li, T. L. New, S/ Dong, Z. M. Oo, "Refine U-Net: Improved U-Net with Progressive Global Feedbacks and Residual Attention Guided Local Refinement for Medical Image Segmentation," *Pattern Recognit. Lett.*, vol. 138, pp. 267-275, 2020, DOI: 10.1016/j.patrec.2020.07.013.
10. K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," arXiv:1512.03385, DOI: 10.48550/arXiv.1512.03385
11. A. Chen and C. Asawa, "Going Beyond the Bounding Box with Semantic Segmentation," *The Gradient*, May 11, 2018, https://thegradient.pub/semantic-segmentation.
12. O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," arXiv:1505.04597, DOI: .48550/arXiv.1505.04597
13. V. Khandelwal, "The Architecture and Implementation of VGG-16," *Medium*, 2020, https://pub.towardsai.net/the-architecture-and-implementation-of-vgg-16-b050e5a5920b.
14. N. Tomar, "What is Intersection over Union (IoU) in Object Detection?," *Idiot Developer*, Feb. 2023, https://idiotdeveloper.com/what-is-intersection-over-union-iou/.
15. https://github.com/RichardsuC/DTU-Image-Segmentation-Project