Application of a deep learning semantic segmentation model to helium bubbles and voids in nuclear materials.

S. Agarwal[1], A. Sawant[2], M. Faisal[3], S. E. Copp[1], J.Reyes-Zacarias[1], Yan-Ru Lin[4], S. J. Zinkle[1,4]

[1]Department of Nuclear Engineering, University of Tennessee, Knoxville, TN 37996, USA

[2]Google, Sunnyvale, CA, 94089, USA

[3]Saudi Network Services, Riyadh, 12245, Saudi Arabia

[4]Materials Science & Technology Division, Oak Ridge National Lab, Oak Ridge, TN 37831, USA

Abstract

Imaging nanoscale radiation-induced defects using the transmission electron microscope (TEM) is a key factor in the successful implementation of materials for nuclear energy structural applications. Analyzing each defect in a TEM micrograph is currently a manual task. To identify the defects in a single image can take anywhere from 15 minutes to an hour and a project can require the analysis of anywhere from tens to ≥ 100 images. Here, we use artificial intelligence (AI) models to automate this task. For simplification, we evaluated images with only a single type of defect; helium bubbles. We performed semantic segmentation of these helium bubble defects in electron microscopy images of irradiated FeCrAl alloys using a deep learning DefectSegNet model. This model, which was previously used to classify crystal defects, is inspired by the classic DenseNet and U-Net image segmentation models. It claims high spatial resolution, but has poor performance at object boundaries. Our paper improves the DefectSegNet model's application by adding two new features. First, the DefectSegNet model is applied not only to perform calculation not pixel-wise but also object (or feature) wise. Because object-wise metrics are directly relevant to our final goal of detecting bubbles, whereas pixel-wise classification is only an intermediate step, it is an important part of our study. Second, a distance map loss (DML) function has been added to increase its performance at object boundaries. It is crucial to accurately represent defects boundaries, especially bubbles, in order to track the bubble-induced swelling caused by irradiation. The boundary-focused DML function is also compared to other loss functions like Cross-entropy, Weighted Binary Cross Entropy (WBCE), Dice and Intersection over Union (IOU). Finally, by incorporating new features, we found a marked improvement on segmentation quality and better shape preservation at the boundaries and areas of the bubbles.

1. Introduction

Helium is an unavoidable by-product in nuclear reactors. In fission reactors, it is due to emission of α-particles from (n, alpha) transmutation reactions occurring in structural materials of current fission reactors [1], [2]. In fusion reactors, it's due to tritium decay and (n, alpha) reactions [3]. Helium is essentially insoluble in solids and diffuses readily, leading to formation of cavities or helium-filled bubbles which cause swelling and embrittlement of the material. The presence of helium cavities on grain boundaries of a material can cause grain boundary cavitation, which can lead to the formation of

intergranular channels that could serve as routes for releasing radioactive elements to the environment. Therefore, the safe deployment and operation of nuclear reactors depend strongly on the study of helium behavior within nuclear materials.

The interaction of helium with the vacancies of materials results in a single or two-atom He-V (helium–vacancy) complex. These He-V complexes interact and when pressure due to buildup of He atoms is equal or greater than the surface tension, these complexes are known as bubbles. Under certain scenarios such as high temperatures, helium atoms can be released. In another situation, there might be a strong influx of vacancies. Both scenarios lead to a decrease of pressure in bubbles and such He-V complexes are also known as voids [4], [5]. The sizes of helium bubbles in most materials is generally less than 10 nanometers (nm) whereas voids can be 50 nm or larger. In general, bubbles and voids can be distinguished from one another based upon their shapes. The shape of bubbles is generally spherical while those of voids are more polygonal. Each complex's He atoms/vacancy ratio further confirms their nature. Both the bubbles and voids could be simultaneously present in the material. One of the most important steps in studying helium behavior is to image the nanoscale cavities or helium bubbles using a TEM microscope and post-analyze TEM images that contain a few hundred bubbles. The post-analysis of TEM images is done to extract information and measurements about the helium bubbles, such as their size, shape, distribution, and volume that can be used to calculate the swelling of material. To be certain that a material can be safely used in the reactor, it is crucial to know the level of swelling that has been experienced by the material.

In TEM micrographs, helium bubbles and voids are imaged using Fresnel contrast. While the TEM images show the nano features precisely, the images still have a lot of errors. For example, the voids and bubbles can be overlapping in 2D projections. Also, the boundaries are not always clear. Moreover, the bubbles appear white and black during under focus and over focus conditions, respectively. This can lead to different sizes when used with large under focus and over focus values. Analyzing each defect in a TEM micrograph is currently a time-consuming manual task. To identify the defects in a single image can take anywhere from 15 minutes to an hour and a project can require the analysis of anywhere from tens to ≥ 100 images. Manual quantification of the images creates a significant obstacle in the quantification of the structural degradation occurring in components. Similar to Anderson et. al. [6], we also consider the following three main problems with manual identification of bubbles [6]. First, the process is time consuming: an individual image can take up to a few hours to classify. Second, manual identification is error-prone as bubbles can be easily misidentified. Third, there is a lack of reproducibility and consistency from one human inspection to another. To address the above, the main objective of this study is to accurately segment the shape and boundary of helium bubbles in transmission electron microscopy (TEM) images. While it is important to detect the number of helium bubbles, accurately segmenting their shape and boundary is challenging due to overlapping bubbles and unclear boundaries. The accurate detection of bubbles leads to an accurate density measurement value, whereas accurate segmentation of helium bubbles gives an exact value for the area covered by each bubble, which can be directly translated into the volume occupied by helium bubbles and the swelling induced by them. The swelling is indeed the multiplication of density into the volume of each bubble. Accurate swelling evaluation is crucial for assessing nuclear materials' suitability in reactors.

Our research aims to automate the identification and accurate segment of TEM images containing helium bubbles and voids using artificial intelligence. This is achieved through the use of deep learning models that analyze patterns in images, which can be broken down into two methods: (i) object

detection and localization, and (ii) image segmentation. The object detection and localization method locate all regions of interest (ROIs) in the image by using a bounding box and then follows up with classification. Image segmentation, which is a pixel-wise classification, allows the entire image to be split into several classes or ROIs. A digital image can be divided into several subgroups called image segments, and a label is assigned to each pixel so that pixels with the same label share certain characteristics. Finally, an image segmentation model generates the mask that only includes pixels corresponding to each ROI.

The biomedical application provides a significant amount of literature that has analyzed objects similar in appearance to helium bubbles and voids and uses both object detection and image segmentation methods for different use cases. In material science application, Jacob et al. [7], recently reviewed the use of deep learning-based object detection and segmentation models in the analysis of features in electron microscopy images. They discuss recent studies that have used different object detection and segmentation models to characterize and quantify defects, particularly dislocation loops in metal alloys [8]**.** The authors also review the work of Anderson et al.[6], who conducted the only known investigation of helium bubbles using any deep learning models and the various challenges encountered during such studies.

Anderson et al. [6] used Faster R–CNN, an object detection model that generated the bounding boxes containing the helium bubbles. Object detection models are easier to use, have better run-time performance, and offer faster training, making them the preferred choice for detecting objects rather than pixel-wise classification codes (or image segmentation models). For cases such as those analyzed in [6], where the helium bubbles are mostly spherical, object detection algorithms that create bounding boxes are an ideal choice. However, the downside to object detection algorithms is their inability to segment image pixel wise (or produce a precise mask) that contains information about the shape of the bubbles. They are only useful in cases where the bounding box dimensions can be directly converted to object dimensions, such as the radius in the case of helium bubbles that have perfect spherical shape. The bounding box that surrounds a non-spherical object does not provide any information about its shape. In real-world scenarios, spherical, elongated, polygonal shaped under-pressure bubbles (or voids) and irregular shape (particularly on grain boundaries) co-exist. Therefore, for TEM micrographs that contain different shapes of bubbles, image segmentation algorithms that provide masks are generally more preferable for helium bubble detection than object detection codes, making them useful for accurately analyzing the distribution and area (or volume) of helium bubbles and voids in irradiated materials which is crucial for understanding the degree of swelling in the material. This is a critical factor that affects the material's structural integrity and longevity in various applications.

There are different algorithms that have used different deep neural network (CNN) architectures for semantic image segmentation. CNNs applications are very challenging because they reduce feature map resolution while performing sub-sampling and pooling operations, which results in a loss of spatial information that leads to poor boundary separation, and thus compromises segmentation accuracy. To solve this, more advanced CNN models such as Fully Convolutional Networks (FCNs)[9] , DeconvNet [10], U-Net [11] or Seg-Net [12] were developed. The architectures of these models are similar, but there are some differences. In recent years, researchers have also started rethinking segmentation by exploring transformer-based models and novel strategies like pixel-query, which leverage attention mechanisms to capture relationships within the input image [13]. In our use case, we selected U-Net among various image segmentation models because of its accuracy and efficiency in terms of both

memory and computational times and also it has been applied to datasets similar to this paper but mostly in biomedical applications [11]. We have used the U-Net based DefectSegNet code [14] model in our study to segment TEM micrographs containing helium bubbles and voids and provide masks.

The paper is structured as follows: Section 2 discusses the experimental setup, including the acquisition and labeling of images and data augmentation. In Section 3, we describe the proposed methods along with the evaluation metric used. Section 4 presents the results and discussions. Finally, Section 5 provides a conclusion.

2. Experimental Setup

For image segmentation models to be developed and trained effectively, it is important that data are correctly acquired, selected and accurately labeled.

2.1. Details on image acquisition

The TEM images were provided by the co-author Yan-Ru Lin who acquired these images as a part of his thesis work at the University of Tennessee. His work involved the study of Fe-Cr ferritic-martensitic (FM) steel, a promising nuclear material candidate for fusion and advanced fission reactors. His research was mainly focused to determine the effects of solutes and irradiation temperatures on cavity swelling under ion-versus neutron irradiation conditions. The TEM images used in this work were taken on a variety of FM steel samples with different Cr concentrations. The samples were simultaneously irradiated using ion and gas (8 MeV Ni^{3+} ions and energy degraded 3.5 MeV He^{2+} ions), in order to quantify the cavity swelling behavior. The irradiations were performed at a temperature range (400-550 °C) with a mid-range dose of ~30 displacements per atom (dpa) and 0.1 appm/dpa He implantation rate. The related publications [15], [16] contain more information on TEM microscopy as well as the technique used to image them.

2.2. Test-train split and image labeling

At the start of the project we pseudo-randomly split the images into two sets. The training dataset contains 108 raw images and the test dataset contains 28 raw images. Both test and training images were contrast enhanced using OpenCV2 implementation of Contrast Limited Adaptive Histogram Equalization (CLAHE) [17]. In the next step, the images were manually labeled to be used to train the segmentation model. Many free labeling tools are available to assist with this task. In our case, VGG Image Annotator (VIA) was chosen [18] . VIA is a single HTML file which can be opened in any web browser without the need to install anything. The user interface is intuitive and friendly. Once images have been imported, users simply need to choose the area shape (ellipse, circle) or mark points (polygon), surrounding objects such as voids and bubbles in this case. All the information can be saved to a JSON (JavaScript Object Notation) file which is a standard format.

2.3. Data augmentation

Data augmentation enhances dataset diversity and size by applying various image processing transformations, such as rotations, translations, crops, mirror effects, Gaussian noise, and contrast enhancements. This technique aids in reducing overfitting and improving model generalization in scientific research, especially with limited data. In our study, we followed the data preparation methodology as in [14], rotating training images by 90, 180, and 270 degrees, resulting in 432 images.

We mirrored these images, expanding the dataset to 864 images. Adhering to the same data preparation methodology as in [14] enabled a fair performance comparison between their model and ours. Adopting these data augmentation techniques aimed to improve the model's performance and robustness when handling diverse image data within a scientific context while facilitating a direct comparison between the models in a consistent manner.

3. Proposed Method

Semantic segmentation refers to the process of associating every pixel in an image with a specific class label. As we have only two classes, the term binary segmentation would be used to describe our study. The pixels in an image would either belong to class I (helium bubbles and voids) or class 2 (background). We used the architecture of DefectSegNet model [14] as a basis for semantic segmentation, inspired by the U-Net and DenseNet models, and we applied improved loss functions in order to enhance the shapes and boundaries of the object detected.

In general, a segmentation model has 2 two key parts - an image encoder and a decoder. The encoder down samples the image so we can analyze the image in a computationally efficient manner, though the tradeoff is loss of information. The Decoder up samples the image by using the various numerical outputs of the encoder layers to map features back onto a coherent image. In order to preserve more of the information otherwise lost during the encoding process, U-Net models allow to directly connect the encoder and decoder via connections called "skip connections". DenseNet has dense “blocks” within which all layers with matching feature-map sizes connect directly with each other and also incorporates skip connections both within and across these blocks. The main advantage of using the DefectSegNet model is that it builds on two models (U-Net [11] and DenseNet [19]); both models are specialized in preventing loss of spatial information. This means it has improved capability of detecting very small and complex features with high accuracy, thus claiming high spatial resolution.

Our method improves the DefectSegNet model's application used in [14] by adding two new parameters. First, the DefectSegNet model is used not only to perform classification pixel-wise, but also object-wise. In object-wise classification, the group of pixels that covers the object (such as void in this case) is compared to the group of pixels covering the object in ground truth. However, in pixel-wise classification, the predicted image is compared with the ground truth image at per pixel level. Because object-wise metrics are directly relevant to our final goal of detecting bubbles, whereas pixel-wise classification is only an intermediate step, it is a crucial part of our study. Second, a distance map loss function has been added to increase its performance at object boundaries. It is important to accurately represent defects boundaries, especially for bubbles, in order to track the bubble-induced swelling caused by irradiation. The boundary-focused distance map loss function is also compared to other loss functions like Cross-entropy, Weighted Binary Cross Entropy (WBCE), Dice and Intersection over Union (IOU). As mentioned earlier, we used the same DefectSegNet architecture as in [14]. The details on different loss functions are described in the next section. The hyperparameters used during the model training are shown in Table 1. It is important to mention that we used the Adam optimizer together with exponential decaying learning rate [20].

Table 1: Hyper-parameters used during the model training for the DefectSegNet code.

Parameter	**Value**
Learning rate	0.001 exponentially decaying
Mini-batch size	32
Epochs	50
Optimizer	ADAM

3.1. Loss functions

To train any neural network, a loss function is minimized in order to improve its accuracy and speed. The loss function gives the model a notion of how far the prediction is from the desirable output and gives feedback to the model while it is training in terms of weight updates, particularly if it is supervised like in this example. In a supervised model, the model learns from pre-labeled data (in our case labeled masks) and guides it to find the best approximation to map input data with output data.

To measure the model's performance from various standpoint, we measured different loss functions based on region, distribution and boundary that are commonly used for image segmentation models.

3.1.1. Distribution Measure based Loss Functions

Binary Cross Entropy loss: The Cross-entropy is used to measure the difference between two probability distributions. The binary cross-entropy (BCE) loss therefore attempts to measure the differences of information content between the actual and predicted image masks. If you consider a target, or underlying probability distribution, P, and an approximation to the target distribution Q then the cross-entropy Q from P is the number of additional bits that represent an event using Q instead. This helps shift the model's output distribution towards the target. The BCE loss examines each pixel individually, comparing the class predictions vector to the ground truth vector. For the case of binary segmentation, let P(Y = 0) = p and P(Y = 1) = 1 - p correspond to ground truth and the respective prediction probabilities are given by $\hat{p}$ and 1- $\hat{p}$, the BCE loss can be defined as:

$$CE(p,\hat{p}) = -(p \log(\hat{p}) + (1-p) \log(1-\hat{p})) \ldots \quad \text{eq. (1)}$$

Weighted Binary Cross Entropy loss: The cross-entropy loss evaluates the class predictions for each vector of pixels individually, and then averages all pixels. This results in equal learning for each pixel. This can lead to problems if there is an imbalance in the representation of different classes within the image. For example, the dominant class may dominate training. In other terms, images with very heavy class imbalance (such as detecting small bubbles from our images; no. of pixels covering the bubbles are highly imbalanced or less than the pixel covering background) may not be adequately evaluated by BCE. Jadon et al. (2020) [21] discussed weighting the cross-entropy loss (WCE) by adding β in order to counteract a class imbalance present in the dataset. WCE was defined as:

$$WBCE(p,\hat{p}) = -(\beta p \log(\hat{p}) + (1-p) \log(1-\hat{p})) \ldots \quad \text{eq. (2)}$$

By using the above formula, the positive examples (masks of the bubbles or voids) get weighted by some coefficient. To decrease the number of false negatives, β is set to a value larger than 1, and to decrease the number of false positives β is set to a value smaller than 1.

3.1.2. Overlap Measure based Loss Functions

Before we go into detail about overlap-based loss function, the brief summary of various evaluation terms presents in any model's confusion matrix output that facilitates the pixel wise calculation such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) is provided. A TP is an outcome where the model correctly predicts the positive class. Similarly, a TN is an outcome where the model correctly predicts the negative class. However, FP and FN, respectively, indicate how many positive or negative samples your model incorrectly predicted. Fig. 1 shows TP, TN, FP and FN for a predicted mask using the DefectSegNet model on our database.

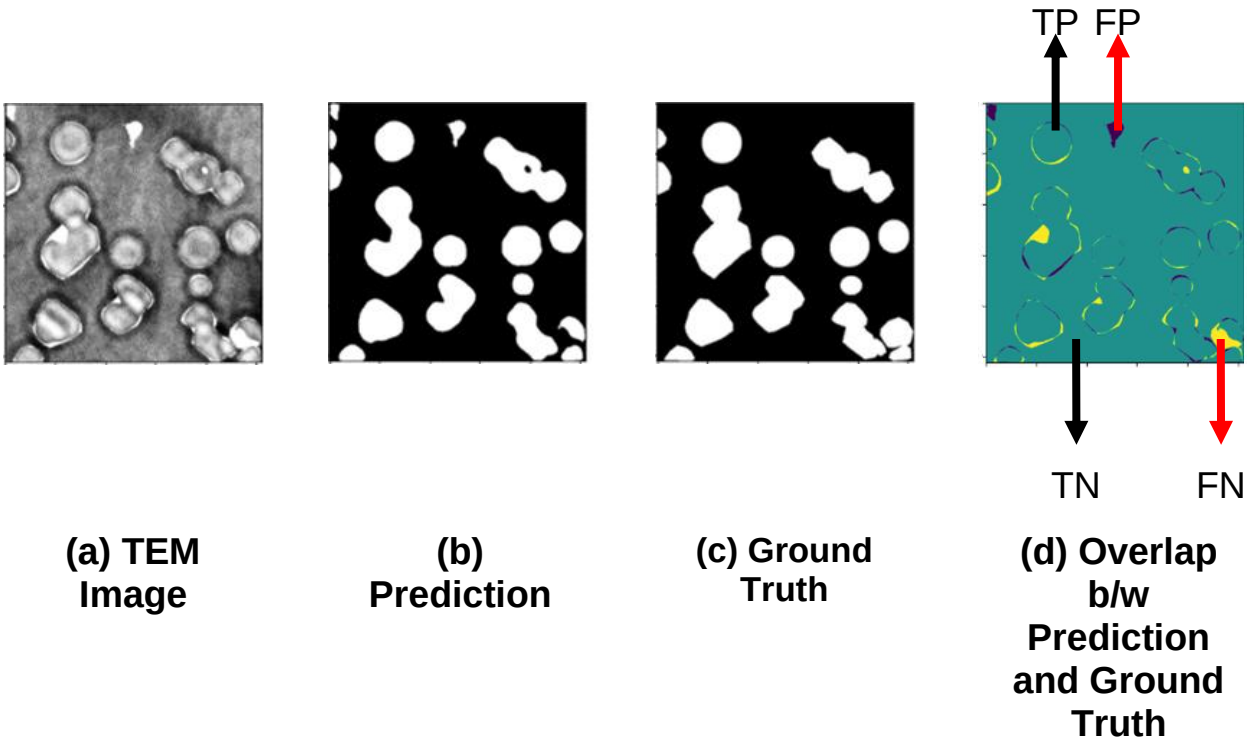

Fig. 1. (a) TEM Image showing helium voids and bubbles (b) DefectSegNet pixel-wise semantic segmentation prediction (referred as masks), of helium bubbles and voids in TEM image. (c) The ground truth labels provided by humans. In (d), an example of the confusion matrix elements: true negative (TN), true positive (TP), false negative (FP), and false negative (FN) is shown.

Dice Loss or F1 score

The Dice coefficient (DC) is a widely used metric to calculate the similarity between two images. The Dice coefficient, which is basically a measure of overlap between two samples, is equivalent to F1 score. This measure can range from 0 to 1. A Dice coefficient of 1 indicates perfect overlap.

$$DC = \frac{2\,TP}{2\,TP+FP+FN} = \frac{2\,|X \cap Y|}{|X| + |Y|} \quad \text{.... eq. (3)}$$

where X and Y are the predicted and ground truth segmentation, respectively.

Continuous version of the Dice coefficient can be used as a loss function. Dice loss (DL) for a pixel is defined as:

$$DL = 1 - \frac{2p\,\hat{p}}{p + \hat{p}} \quad \text{.... eq. (4)}$$

where p and $\hat{p}$ are the same in eq. 1 and 2. F1 score provides a single score that balances both the concerns of precision and recall in one number. F1 score is the harmonic mean of precision and recall, which means that the F1 score tells the model's balanced ability to both capture positive cases (recall) and be accurate with the cases it does capture (precision).

Intersection over Union

Intersection over union (IoU) is a standard metric for semantic image segmentation tasks. For each class, it is a measure of the area of the correctly classified pixels for a particular class, as a fraction of the area of all pixels that were predicted as belonging to that class as well as all pixels actually belonging to that class. This metric is evaluated over the entire dataset for all classes.

The intersection over union (IoU) or the Jaccard metric is computed as:

$$IoU = \frac{TP}{TP+FP+FN} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|} \quad \text{.... eq. (5)}$$

where X, Y, TP, TN, FP, FN are the same as Dice loss. From eq. (3) & (5), it can be seen that DC $\geq$IoU. The IoU is very similar to the Dice coefficient. They are positively correlated, meaning if according to IoU model A is better than model B at segmenting an image, then the DC will say the same. Like the DC, it also ranges from 0 to 1, with 1 signifying the greatest similarity between predicted and truth.

IOU Loss (IL) for a pixel is defined as:

$$IL = 1 - \frac{p\hat{p}}{p + \hat{p} - p\hat{p}} \quad \text{...eq. (6)}$$

3.1.3. Boundary based Loss Function

Boundaries of an object are often hard to segment and models a common source of errors. This motivates the use of distance map-based loss function which encodes boundary information for objects

in the loss function. One common way of achieving this is to weigh points closer to the boundary higher than the points away from the boundary. Following Calivia et al. [22], we define Distance Map Loss (DML) for a pixel as:

$$DML = (1 + \phi)BCE \quad \text{...eq. (7)}$$

where BCE is the binary cross entropy loss and ϕ is derived from the normalized distance map for the image. Its value is 1 on boundaries of an object (void) and 0 on the interior-most points.

3.2. Performance evaluation metrics

To evaluate the performance of object detection and semantic segmentation tasks, several common metrics are used, including accuracy, precision, recall, mean area error, F1-score, IOU, and Dice. In our case, we did not calculate the Dice coefficient as it is highly correlated with the Intersection over Union (IoU) metric; both of these metrics measure the degree of overlap between the predicted segmentation and the ground truth. While we briefly mentioned the F1-score and IOU in the context of their use in the loss function, the other metrics are defined below:

Pixel accuracy

Accuracy refers to the overall correctness of the machine learning model, calculated as the percentage of correctly predicted pixels out of the total number of pixels.

$$Pixel\ accuracy = \frac{TP + TN}{TP+FP+TN+FN} \quad \text{...eq. (8)}$$

Precision

Precision measures the accuracy of the model in predicting a specific category by calculating the ratio of correctly predicted true positive pixels to the total number of positive pixels predicted.

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{...eq. (9)}$$

Recall

Recall counts the number of times the model accurately detects a particular category, calculated as the ratio of accurately predicted true positive pixels to the total relevant pixels of the category. False negatives are penalized as they lead to underestimation.

$$\text{Recall} = \frac{TP}{TP+FN} \quad \text{...eq. (10)}$$

Mean Area Error (MAE) and Absolute Area Error (MAAE)

Mean area error (MAE) and mean absolute area error (MAAE) measure the difference between the predicted and ground truth area of an object, with MAAE measuring the absolute difference.

4. Result and Discussion

4.1. Quantitative comparison of different loss functions

Table 2 displays the precision, recall and accuracy, mean absolute error, and F1-score per pixel calculated for different loss functions. Figure 2 shows the clear comparison of various performance metrics in Table 2. For all loss functions, the accuracy and F1 at the pixel level are nearly equal (or vary within a small range). Accuracy, which is used to measure the correctness of the model, is 0.95 for all loss functions. This accuracy value is a great indicator that the model's overall performance is excellent. The F1-score, which is known to balance precision with recall for positive classes, is approximately (or mostly) 0.87 for all loss functions.

Most loss functions have recall values between 0.85-0.88. However, the "WBCE" loss function stands out with a recall value of 0.93, indicating a significant improvement over the other loss functions. This improvement comes at a cost, though. "WBCE" has a lower precision value and a higher mean absolute area error (0.16) compared to other loss functions, which could be due to the use of an additional coefficient to handle imbalanced data sets. Despite the common belief that "WBCE" should improve the model's performance on imbalanced data sets, our study shows that it did not significantly improve our model's performance. This unexpected result might be due to the bias towards marking negative pixels as positive, which is inherent in "WBCE" because of the extra weight given to the positive class. This bias is reflected in the precision and recall numbers, where the positive class has a higher weight, causing the model to predict more positive pixels, increasing recall but decreasing precision.

For our application, the accuracy of the object area measurement is critical to calculate the swelling of material caused by helium and select safe nuclear material. Therefore, "WBCE" is less useful because it produces a greater area error. In contrast, the DefectSegNet model in [14] used only "WBCE" as the loss function but did not perform well on our dataset. This difference in performance could be due to variations in image properties, defect types, or defect distribution. Given the poor performance of "WBCE" on our dataset, we decided to try other loss functions, particularly distance, to improve the area error and pixel classification at the border.

Table 2: Performance metrics per pixel at the 0.5 threshold for different loss functions.

Loss functions	Precision	Recall	Accuracy	F1-score	IOU	Mean Area Error	Mean Absolute Area Error
IOU	0.843	0.874	0.952	0.858	**0.752**	0.041	0.085
Dice	0.865	0.883	0.958	0.874	**0.776**	0.025	0.077
WBCE	0.805	0.926	0.951	0.861	**0.757**	0.153	0.157
BCE	0.871	0.864	0.956	0.868	**0.766**	-0.006	0.063
Distance	0.881	0.850	0.956	0.865	**0.763**	-0.032	0.072

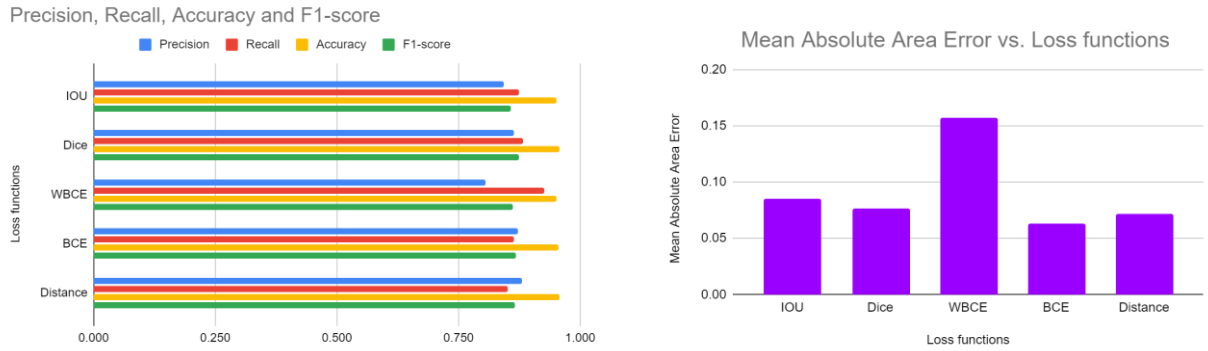

Fig. 2. Charts for different performance metrics (per pixel) at the 0.5 threshold for different loss functions.

In the typical approach, models are evaluated by looking at their accuracy, precision, and recall for each pixel individually. However, in the case of detecting helium cavities in materials, it will also be beneficial to evaluate the model's performance on the whole objects rather than individual pixels. To address this, we compared the model's pixel-wise prediction to the ground truth using IoU to identify TP (true positive), FP (false positive), and FN (false negative) at the feature (or object) level. We used this comparison to generate precision, recall, and F1 scores at IoU thresholds of 0.5, which is the minimum overlap needed for a feature to be considered a TP. This approach is also referred to as object-wise metrics in this paper. It is just an additional metric to other classic metrics that we are reporting. We took this idea from object detection models such as Faster R-CNN, for example Anderson et al. [6] applied this approach on the bounding boxes using Faster R-CNN, and we thought of extending this to our study and applying it to objects. Object-wise metrics are more relevant to our objective of detecting voids, while pixel-wise classification is only an intermediate step. Therefore, these metrics are more significant to our overall objective. Table 3 presents the precision, recall, accuracy, and F1-score that have been calculated object-wise for different loss functions. Figure 3 provides a clear comparison of the various performance metrics in Table 3.

Among the performance metrics shown in Table 3, we can see that the best-performing loss function in terms of object-wise performance metrics is Dice, with the highest values for precision, recall, accuracy, IOU, and F1-score. This indicates that the Dice loss function may be the most effective for detecting helium cavities in materials at the object level. “Dice” has the highest values for “Precision” (0.885), “Recall” (0.830), “IOU” (0.793), and “F1” (0.856) and all other loss functions perform the same.

Based on the table provided, the DML ranked second to Dice and achieved a precision of 0.848, recall of 0.810, accuracy of 0.712, IOU of 0.736, and F1 score of 0.828, outperforming all other evaluated loss functions except for the Dice loss function.

DML is a distance-based loss function that combines BCE with the concept of distance maps to better capture object boundaries and improve object detection performance. DML is defined as $(1 + \alpha) * BCE$, where α is derived from the normalized distance map for the image. Its value is 1 on the boundaries of an object (void) and 0 on the interior-most points. The BCE loss function is typically used to measure the similarity between the predicted object and the ground truth object at the pixel level. It is applied independently to each pixel, penalizing the model for making incorrect predictions, with the goal of minimizing the difference between the predicted probability distribution and the true binary labels. However, it does not consider the distance between the predicted and true object boundaries, which can result in suboptimal performance, particularly when objects have complex or irregular shapes. By combining BCE and the distance map, the DML function considers the distance between the predicted and true object boundaries and penalizes incorrect predictions more heavily near the boundaries. This improves the model's ability to capture object boundaries accurately, leading to better object detection performance. In summary, combining BCE and distance map into the DML function has led to better object detection performance for our dataset by incorporating shape information into the loss function and penalizing incorrect predictions near object boundaries more heavily.

In terms of potential future directions, combining distance-based loss functions with the Dice loss function could be beneficial. The Dice loss function measures the overlap between the predicted object and the ground truth object, and by combining the two, the loss function can potentially incorporate both shape and overlap information, leading to even better object detection performance. This approach would be particularly useful in cases such as this study where the object shapes are irregular or complex. The qualitative comparison in terms of actual segmentation images will be discussed in the next section.

In Table 3, the worst-performing loss functions in terms of object-wise performance metrics are "WBCE" and "BCE", which have the lowest values for precision, recall, accuracy, IOU, and F1-score. The WBCE loss function performs worse than other loss functions such as Dice, BCE, and Distance, which have higher values for these metrics. WBCE also showed similar results for pixel-wise classification. It appears that Cross Entropy based loss functions work best for balanced datasets, and introducing a weighting coefficient seems like an artificial way to correct imbalanced data. This suggests that the WBCE loss function may not be the most effective for segmenting images with datasets similar to ours.

Table 3. Performance metrics per object at the 0.5 threshold for different loss functions.

Loss functions	Precision	Recall	Accuracy	IOU	F1
IOU	0.850	0.788	0.692	**0.740**	0.818
Dice	0.885	0.830	0.752	**0.793**	0.856

WBCE	0.840	0.780	0.681	**0.724**	0.809
BCE	0.835	0.781	0.681	**0.716**	0.807
Distance	0.848	0.810	0.712	**0.736**	0.828

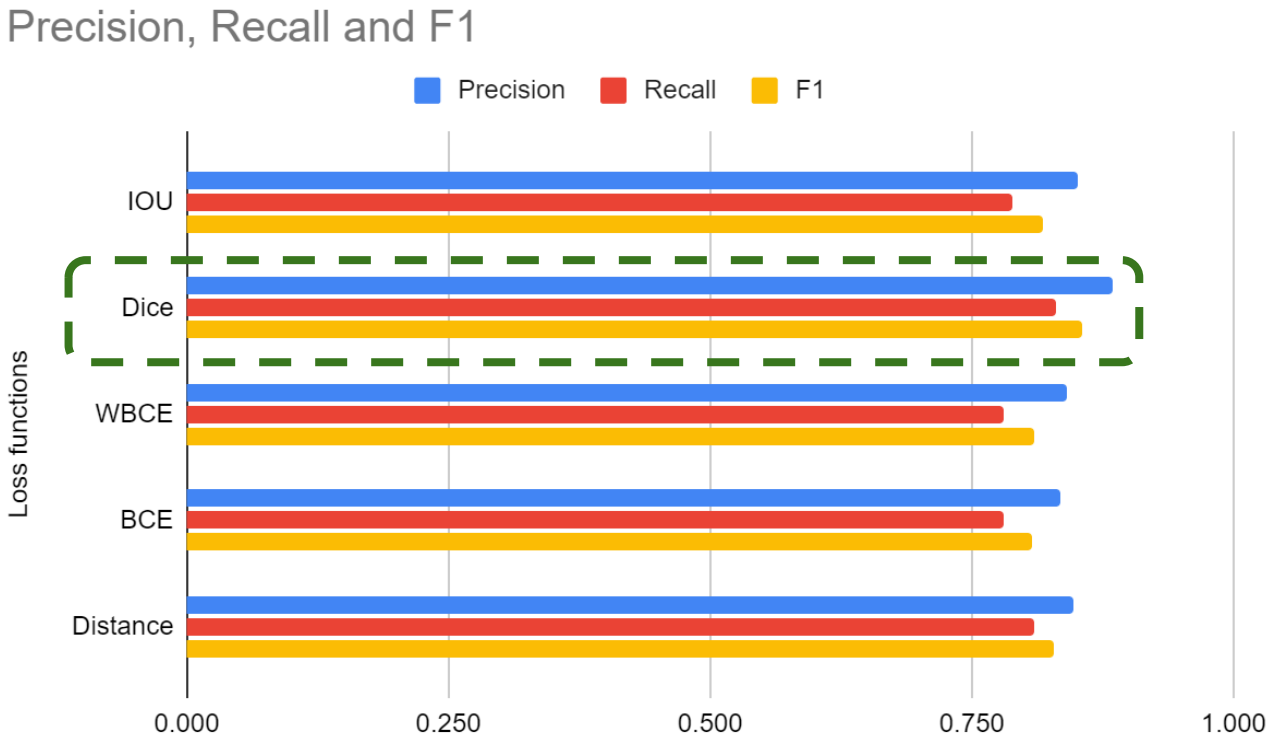

Fig. 3. Charts for different performance metrics (per object) at the 0.5 threshold for different loss functions.

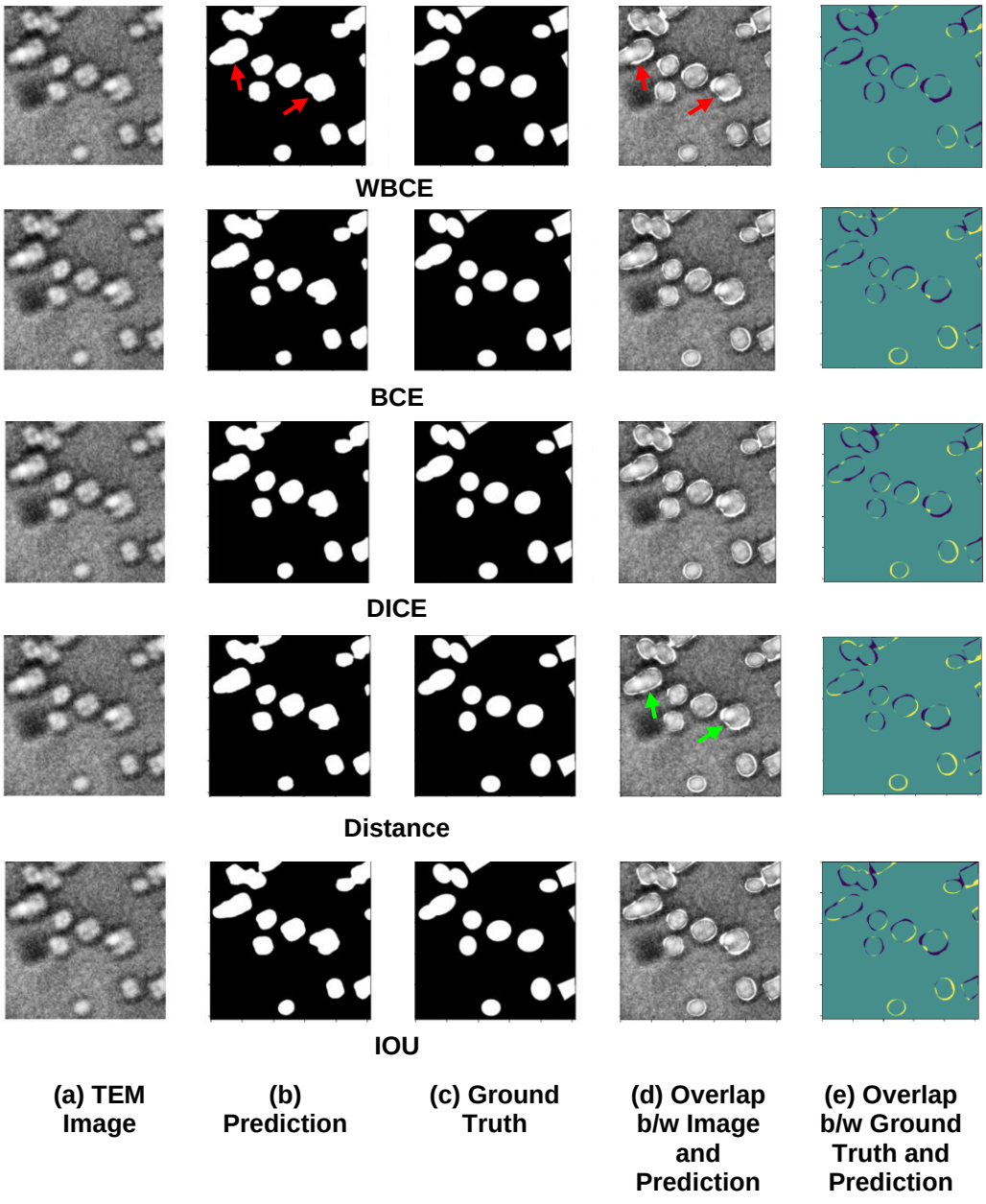

Fig. 4. The image segmentation results using different loss functions (a) The TEM image of the helium bubble and voids. (b) DefectSegNet pixel-wise semantic segmentation prediction (referred as masks), of helium bubbles and voids using TEM images in (a). (c) The ground truth labels done by humans. (d) The overlap of DefectSegNet prediction maps with the original TEM image in (a). (e) The overlap of DefectSegNet prediction maps with the Ground Truth labels done by humans; dark blue is false positive, yellow is false negative and teal is where Prediction matches with mask.

4.2. Qualitative comparison of different loss functions

Fig. 4 shows that all loss functions, except WBCE, have produced a similar quality of segmentation. WBCE's case (see marked bubbles by red arrow in the first row (d)) the masks overlap slightly more than TEM images when they are used with other loss functions. In some cases, WBCE has predicted extra white areas (not shown in Fig. 4) which is not predicted in other cases. As mentioned, WBCE is biased towards marking more positives therefore the objects that are composed of white pixels are a bit larger as compared to other loss function and that's why it is also showing much more error in the mean area as compared to all other loss functions (refer to Table 2 for comparing area values).

The masks created by the distance loss function are very close to the real shapes of voids or bubbles, which is not apparent in the performance metrics shown in Table 2 and 3. However, on qualitative comparison, Fig. 4 shows that masks obtained using distance loss function are more precise (less broader shapes and thin boundaries) than other loss functions, especially WBCE (see marked bubble in fourth row (d)). More qualitative comparison of WBCE and DML loss function could be found in supplementary file.

Finally, we found using the DML function that combining binary cross entropy with distance maps results in better segmentation than cross entropy by all measures.

5. Conclusion

In this study, we utilized the DefectSegNet [14] model to perform semantic segmentation. The DefectSegNet model is based on the U-Net and DenseNet image segmentation models. However, the original model in [14] used only one loss function, specifically WBCE. This study attempted to replicate this approach, including data augmentation techniques and architecture, but with a different dataset containing defects such as helium bubbles and voids. However, the same approach did not perform well for our dataset, especially in terms of segmentation quality at the borders. Accurate area and shape calculations of helium voids and bubbles are critical for evaluating the performance of nuclear materials under irradiation in nuclear reactors, hence better segmentation quality at the borders was needed. To address this, additional loss functions, particularly distance, were introduced to improve segmentation quality. Using these additional features, the model in this study outperformed the one mentioned in [14]:

Firstly, we evaluated the segmentation pixel wise as well as object wise. Specifically, after conducting pixel-wise segmentation, we took the output and compared it with the ground truth using an Intersection-over-Union (IoU) algorithm to identify the True and False Positives and Negatives at an object level. The comparison was performed to generate the precision, recall, and F1 scores at IoU thresholds of 0.5 which is accepted as the minimum overlap needed for a feature to be acceptable as a TP. This is important because object-wise metrics directly relate to our ultimate goal of detecting bubbles. Pixel-wise classification is just an intermediate step.

Secondly, to improve the model’s performance at object boundaries, a DML function was added. To track void-induced swelling, it is important to accurately represent defects boundaries. This loss function was compared with other loss functions such as Cross-entropy and WBCE, Dice, and IOU. We found that new features resulted in a significant improvement in segmentation quality, better preservation of shape at the boundaries, and around the bubbles.

Overall, we found that most loss functions perform well, other than Weighted Boundary Cross Entropy (WBCE). This loss function is usually used for unbalanced datasets such as the one we have. But our results show that WBCE led to inaccurate and ill-fitting masks which are much larger than the object targeted. WBCE also leads to higher mean absolute area error and much higher mean area error, which is an important parameter for us as it directly influences the volume we predict and hence the material swelling. The DICE loss function performed the best for accuracy, precision and recall in most cases while on qualitative comparison the distance loss function resulted in accurate shapes with well-fitting boundaries.

Bibliography

[1] P. Trocellier, S. Agarwal, and S. Miro, "A review on helium mobility in inorganic materials," *J. Nucl. Mater.*, vol. 445, no. 1, pp. 128–142, Feb. 2014.

[2] S. Agarwal, P. Trocellier, Y. Serruys, S. Vaubaillon, and S. Miro, "Helium mobility in advanced nuclear ceramics," *Nucl. Instrum. Methods Phys. Res. B*, vol. 327, pp. 117–120, May 2014.

[3] H. Ullmaier, "The influence of helium on the bulk properties of fusion reactor structural materials," *Nucl. Fusion*, vol. 24, no. 8, p. 1039, Aug. 1984.

[4] H. Wiedersich, J. J. Burton, and J. L. Katz, "Effect of mobile helium on void nucleation in materials during irradiation," *J. Nucl. Mater.*, vol. 51, no. 3, pp. 287–301, Jul. 1974.

[5] S.-H. Li, J.-T. Li, and W.-Z. Han, "Radiation-Induced Helium Bubbles in Metals," *Materials* , vol. 12, no. 7, Mar. 2019, doi: 10.3390/ma12071036.

[6] C. M. Anderson, J. Klein, H. Rajakumar, C. D. Judge, and L. K. Béland, "Automated Detection of Helium Bubbles in Irradiated X-750," *Ultramicroscopy*, vol. 217, p. 113068, Oct. 2020.

[7] R. Jacobs, "Deep learning object detection in materials science: Current state and future directions," *Comput. Mater. Sci.*, vol. 211, p. 111527, Aug. 2022.

[8] R. Jacobs *et al.*, "Performance and limitations of deep learning semantic segmentation of multiple defects in transmission electron micrographs," *Cell Reports Physical Science*, vol. 3, p. 100876, May 2022.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 3431–3440.

[10] H. Noh, S. Hong, and B. Han, "Learning Deconvolution Network for Semantic Segmentation," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1520–1528.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015, pp. 234–241.

[12] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[13] T. Zhou, W. Wang, E. Konukoglu, and L. Van Goo, "Rethinking Semantic Segmentation: A Prototype View," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 2572–2583.

[14] G. Roberts, S. Y. Haile, R. Sainju, D. J. Edwards, B. Hutchinson, and Y. Zhu, "Deep Learning for Semantic Segmentation of Defects in Advanced STEM Images of Steels," *Sci. Rep.*, vol. 9, no. 1, pp.

1–12, Sep. 2019.
[15] Y.-R. Lin *et al.*, “Bubble formation in helium-implanted nanostructured ferritic alloys at elevated temperatures,” *Acta Mater.*, vol. 217, p. 117165, Sep. 2021.
[16] Y.-R. Lin, A. Bhattacharya, D. Chen, J.-J. Kai, J. Henry, and S. J. Zinkle, “Temperature-dependent cavity swelling in dual-ion irradiated Fe and Fe-Cr ferritic alloys,” *Acta Mater.*, vol. 207, p. 116660, Apr. 2021.
[17] “Website.” https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.htm
[18] “Visual Geometry Group - University of Oxford.” https://www.robots.ox.ac.uk/~vgg/software/via/ (accessed Oct. 20, 2022).
[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 2261–2269.
[20] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, “The marginal value of adaptive gradient methods in machine learning,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, California, USA, Dec. 2017, pp. 4151–4161.
[21] S. Jadon, “A survey of loss functions for semantic segmentation,” in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Viña del Mar, Oct. 2020. doi: 10.1109/cibcb48159.2020.9277638.
[22] F. Calivá, C. Iriondo, A. M. Martinez, S. Majumdar, and V. Pedoia, “Distance Map Loss Penalty Term for Semantic Segmentation,” *ArXiv*, 2019, Accessed: Oct. 18, 2022. [Online]. Available: https://www.semanticscholar.org/paper/8d1ae1fb2fdaddd58e57f089a1df8c78786510cf