This article has been accepted for publication in IEEE Transactions on Plasma Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPS.2024.3505157

Neural Network-Based Defect Detection On Tomography Images For Quality Control of Inertial Fusion Capsules

D.A. Orozco, M.C. Quinn, B.S. Sammuli, A. Allen, K. Boehm, M. Farrell, C. Kong, M. Ratledge, K. Sequoia,
Inertial Fusion Technology, General Atomics

***Abstract*—Recent headlines of record-breaking fusion energy yields using inertial confinement have led to a growing interest in the metrology processes used to inspect the high-density carbon (HDC) capsules that have achieved these yields. To ensure the highest quality HDC shells, the tomographic reconstructions are reviewed for defects to satisfy a certain standard of quality. Historically, a human expert would visually analyze HDC tomography data searching for defects, which is very time-consuming and labor-intensive. While classical computer vision methods have also been employed, machine-learning methods have demonstrated the ability to analyze vast amounts of data at high speeds and more robustly identify patterns and anomalies that may indicate the presence of previously undetectable defects. Presented here is our novel process, ShellNet, that has been developed to automate this defect detection task on HDC shells using convolutional neural networks; the challenges and limitations of this approach are also discussed. A machine-learning model trained on thousands of expert-labeled defects and tomography images, utilizing a custom labeling tool, is now being used in production. Case studies are presented demonstrating the effectiveness of machine learning in this application, as well as outlines for future research in this area. Overall, machine learning has been shown to be an effective method in enhancing the quality-based selection process of HDC shells, even exceeding the recall performance of human labeled data. These improvements promise to lead to overall improved reliability of HDC materials in inertial fusion technology.**

***Index Terms*—Inertial confinement fusion, Neural networks, X-ray tomography**

I. Motivation and Related Work

INERTIAL confinement fusion (ICF) has recently seen significant gains in energy yields, as the engineering capabilities are catching up to the understanding of the physics. The National Ignition Facility met the Lawson Criteria in 2021 [1] and most recently achieved ignition in 2022 [2].

The high neutron yields obtained in inertial confinement fusion are attributed to several improvements, one of which is the High Density Carbon (HDC) capsule ablator material. These HDC capsules are roughly 2mm in diameter with an average 80 micron wall thickness, leaving a strong shell to hold deuterium and tritium fuel. The shells are manufactured through a series of deposition, polishing, and mandrel leaching steps, followed by a number of metrology steps to confirm the quality of the capsules after a batch of shells is fabricated. Prior to using a shell for fusion experiments, a series of radiographs are taken of the shell and converted to tomographic slices using X-ray computed tomography techniques. This metrology step limits characterization to one shell every 16 hours, due to the time required to create the series of radiographs.

Historically, humans would review the slice images created from the tomographic reconstruction to locate defects. This method is extremely laborious and prone to poor detection, especially considering that the tomographic analysis of each shell contains thousands of individual high-resolution images. These limitations in manual defect detection have motivated the exploration of automated defect detection systems to improve efficiency and accuracy in this process.

Classical computer vision techniques are typically hand-crafted algorithms designed by experts in the field based on mathematical models and heuristics. Examples include filtering, edge detection, and feature extraction. A classical computer vision model was first implemented to automate and standardize the defect detection task [3], but this method has its own limitations. Traditional computer vision models are finely tuned for extracting specifically known defect types and do not perform well with visual data drift.

On the other hand, Convolutional Neural Networks (CNN) have had major success in producing high-performing object detection models. They have recently been applied in other inertial fusion related research for defect detection in plastic mandrel [4] and classifying types of damage on optics glass at the National Ignition Facility [5]. Our work presents a method to automate defect detection for HDC shells utilizing a CNN. We refer to this defect analysis pipeline as ShellNet, consisting of image processing, object detection, and automating the continuous integration (CI) and deployment (CD) pipeline.

II. Methodology

Given the challenges of manual detection and computer vision techniques, a data-driven approach using a neural network is implemented to take advantage of expert-analyzed data. This paper describes the object detection process, the challenges of creating ground-truth labels, and the post-detection tools

The methodologies described in parts A-E of Section II and all of Section III have been supported by the U.S. Department of Energy, under the U.S. DOE contract 89233119CNA000063 and their application towards confinement fusion energy technology described in parts F-G of Section II are funded by General Atomics' Internal Research and Development Funding.

necessary to improve the overall defect detection process. The workflow for integrating the detection model into production is described as well as a statistical evaluation of the model's performance on a validation set of data.

A. Object Detection

Given the objective of automatically finding the location and approximate size of defects on HDC shells, object detection naturally lends itself to this task. In this paper, we have chosen to implement RetinaNet, a single-stage neural network-based object detector that has not only showcased state-of-the-art performance but also boasts enhanced speed compared to many of its two-stage counterparts [6].

RetinaNet is a single-stage object detector neural network consisting of a pre-trained ResNet [7] front-end architecture with a Feature Pyramid Network (FPN) backbone. The network accomplishes the two sub-tasks of object detection, classification and localization, by using two independent sub networks.

1) Classification Task: The predicted posterior probability of a defect existing in a particular image, P_i, is described as follows:

$$P_i = \begin{cases} P\left(class = foreground|X_i\right) & if foreground \\ 1 - P\left(class = foreground|X_i\right) & otherwise \end{cases} \tag{1}$$

where X_i is a given input image for image classification, or a given bounding box for defect detection. The corresponding negative log-likelihood (NLL) loss function is derived using maximum likelihood estimation as follows:

$$NLL = -\sum_{i=1}^{N} \ln(p_i) \tag{2}$$

Employing Eq. 1 and Eq. 2 in practice means that we are training our neural network to predict the probability of a defect existing in a given bounding box, thereby optimizing our model weights to minimize the loss in Eq. 2 over the N total predicted bounding boxes.

The standard binary cross-entropy (BCE) loss shown in Eq. 2 commonly suffers from foreground-background class imbalance. HDC shells have an extreme imbalance due to the sparse number of small defects, typically covering less than a thousandth of a percent of an image. One common solution is balanced cross-entropy:

$$BCE = \begin{cases} -\sum_{i=1}^{N} \alpha \ln(p_i) & if foreground \\ -\sum_{i=1}^{N} (1-\alpha) \ln(p_i) & otherwise \end{cases} \tag{3}$$

This loss becomes a modified version of binary cross-entropy that balances positive and negative examples when the α term is set to the inverse frequency of the foreground class. RetinaNet further improves upon balanced cross-entropy by introducing a novel focal-loss (FL):

$$FL = -\sum_{i=1}^{N} \alpha_i (1 - p_i)^{\gamma} \ln(p_i) \tag{4}$$

where α originates from Eq. 3, and the scaling factor, γ, is chosen through hyperparameter tuning. Focal loss introduces the novel scaling term $(1-p_i)^{\gamma}$, which has been demonstrated to down-weight easy examples in the loss function while emphasizing hard negatives [6]. Figures in the original RetinaNet work demonstrate how changing γ controls the degree to which hard vs easy examples are emphasized, with increased γ concentrating the loss on hard examples.

2) Bounding Box Regression Task: The loss function for regression does not suffer from class imbalance because the loss only exists when there are predictable foreground bounding boxes. The output vector representing each bounding box prediction, B^i can be described as $B^i = \left({B_{x_c}}^i, {B_{y_c}}^i, {B_w}^i, {B_h}^i\right)$, corresponding to box CenterX, CenterY, width, and height respectively. We use the corresponding standard regularized least squares solution proposed in R-CNN [8] to optimize our regression network.

B. Expert-labeling of HDC shells

A dataset comprising of 27 HDC shells and approximately 52,000 images was randomly selected from a historical database of HDC tomography reconstructions for expert hand labeling. We utilize two labeling methods for collecting and verifying data, along with introducing a custom "exploitation" search method to efficiently expand ground-truth data.

1) Data Labeling: The first data labeling approach, the "needle in a haystack", utilizes manual human effort by sorting through entire image stacks of a single shell to find and annotate individual defects. This method can take 3 to 4 hours, and proved to be highly time-intensive for minimal gains in defect detection and is typically only used for thoroughly validating model performance against a human. The second approach, "assisted labeling", utilizes preexisting defect proposals (such as a classical computer vision or machine-learning models) to propose regions with higher probabilities of containing defects for human review. A custom labeling tool was developed to provide experts with the capability to efficiently verify, classify, and draw bounding boxes around defects. Integrated into this tool is the "assisted labeling" approach to rapidly generate annotated data-sets.

2) Custom "Exploitation" Search: Although our "assisted labeling" data tool efficiently finds defect instances in independent tomography slices, we further take advantages of the 3D properties of tomographic data. Each 2D tomography image is spatially correlated, potentially spanning several adjacent tomography slices. This means that if a defect can be found in one location on slice N, there is a high likelihood of finding that defect in adjacent slices N-1 and N+1, i.e. P(defect in slice N-1| defect in slice N) = P(defect in slice N+1 | defect in slice N) >> P(defect in random slice). Conversely, knowing a tomography slice N contains no defect, gives us no additional information whether a defect exists in slice N+1, equivalent to picking a spot at random (needle in a haystack). This can be expressed as P(defect in slice N+1 | defect not in slice N) = P(defect in random slice). From these posterior probabilities, a very simple "exploitation search" algorithm is illustrated in Figure 1 and summarized as: If a defect is found in slice

This article has been accepted for publication in IEEE Transactions on Plasma Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPS.2024.3505157

N, human verifies the slices directly adjacent to it. Continue searching adjacent slices until no defect instances are found.

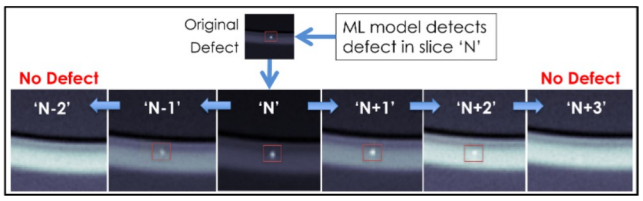

Fig. 1. When a defect is found by the ML model in a given slice, the "Exploitation Search" method is used to find different portions of the defect in adjacent tomography slices. All contiguous images are manually verified until the defect is no longer visible.

When a defect is found by the ML model in a given slice, the "Exploitation Search" method is used to find different portions of the defect in adjacent tomography slices. All contiguous images are manually verified until the defect is no longer visible.

The "exploitation search" quickly expanded ground-truth data and reduces the chance of missing defect instances in adjacent slices. This process is much faster than the "needle in a haystack" approach, on average taking 15-45 minutes per shell, depending on the number of defects. This method is typically used for expanding training dataset size and for semiannual quality checks. To combat any bias that may be introduced by this process, the performance of the model is verified against the "needle in a haystack" approach on a subset of test data.

C. Data Acquisition & Processing

The tomographic datasets were collected using a Zeiss Xradia Versa 510 on 4× magnification with 2,000 x 2,000 x 2,000 voxel count with an average 1.22 microns per voxel conversion ratio for every shell. The volumetric data is converted into a stack of Tiff images representing the cross-sectional slices of the shell. The stack images are processed for defects and used for ground-truth labeling.

1) Tiling: Previous work has shown that object detection models may show high performance on datasets with large pixel coverage, yet yield significantly lower performance of small object detection tasks on high resolution images. To combat this, A recent tiling approach has shown significant boosts in performance on small object datasets while also lowering computation and memory requirements [9]. Considering that HDC shell images are high resolution with defect bounding boxes spanning a pixel coverage lower than even most small object datasets, our implementation of this approach has been crucial.

Most detectable defects found on slices are only a few microns in diameter, with the smallest threshold for detectable defects being a dozen occupied pixels. When compared to an image size of 2,000 by 2,000 pixels, this will lead to a disproportionate background to defect ratio of 600,000 to 1, causing several issues with both training the neural network and inference of datasets. To manage this issue, all image slices are divided into sets of overlapping tiles to decrease the

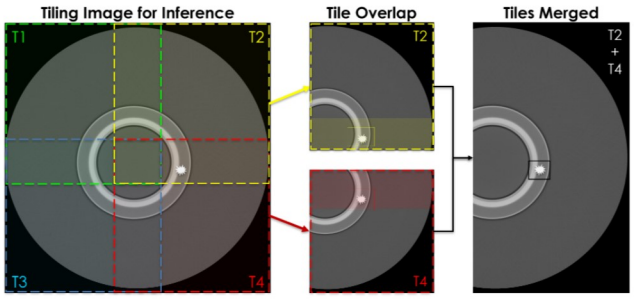

Fig. 2. Tomography image slices are tiled into separate overlapping regions before model inference. Overlap of tiles T2 and T4 are shaded yellow and red, respectively. Defects found in overlapping regions of the tiles are recombined into single defect instances.

imbalance of background to feature ratio as demonstrated by Figure 2. The chosen tile size and percent overlap becomes a hyperparameter to be optimized. To reduce the combinatorial search during hyperparameter optimization, values are randomly sampled from a predetermined list of proposed tile sizes using a multinomial distribution.

2) Image Scaling: Once images are tiled, they are scaled to a fixed aspect ratio and size (no image skewing nor warping). One purpose for scaling is that while model performance may benefit from smaller windows (e.g. 400 by 400 pixel tile size), this may be below the model architectures minimum input size and require resizing. A second benefit observed from image upscaling was that an increased minimum pixel count for extremely tiny defects increased the detection rate of the object detector. This created a trade off on model performance between the benefit of increased minimum defect pixel count (minimum bounding box size) and pixelation penalty caused by stretched resolution. Thus, the scaling factor was treated as a hyperparameter and the final image resolution was found through hyperparameter search.

3) Data Augmentation: Data augmentation increases the diversity and variability of training data and is an important part for improving model performance and generalization. Before normalizing the input training data various image manipulation techniques were employed for data augmentation. Among these, horizontal, vertical and rotational image flipping was used to exploit the fact that the model should be able to detect defects for any given shell orientation. Image contrast and brightness changes were also used as augmentation to simulate the random variations that can result from tomography image settings.

D. Network Hyperparameter Tuning

Identifying an optimal set of hyperparameters is important for the performance and robustness of machine-learning models and their integration into high-quality production assessment systems. These hyperparameters are predetermined variables that shape model and training characteristics, including the model architecture and specifics of the training process (such as learning rate, regularization, and data augmentation/processing strategies). The final model structure and

This article has been accepted for publication in IEEE Transactions on Plasma Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPS.2024.3505157

hyperparameters were chosen experimentally through optimization of various predefined model architectures and image tiling methods (described in section 2C). Parallel tuning of hyperparameters was done using the open-source DeepHyper [10] Python package developed by Argonne National Lab and their implementation of the Centralized Bayesian Optimization search (CBO) algorithm. The final specifications for the RetinaNet model were chosen by best performance and are shown in Table I. The number of hidden layers specified in the table refers to the depth of the ResNet [7] backbone component of RetinaNet [6]. This parameter is chosen prior to training by sampling from a discrete set of pre-trained ResNet models with various hidden layers (e.g. 18, 50, 101, 152, etc.).

TABLE I
FINAL HYPERPARAMETER TUNING VALUES

Hyperparameter	Final Value
ResNet Hidden Layers	101
Tile Overlap	150 pixels
Image Scaling	980x980 $pixels^2$
Optimizer	Adam
Initial Learning Rate	$\approx 12e^{-6}$
Tile Size	600x600 $pixels^2$ (16 tiles)
Batch Size	8
Weight Initialization	Kaiming Normal

E. Model Test Performance

To evaluate the performance of the neural network model, a set of 26 expert-labeled shells are partitioned by shell into a training and validation set. To avoid data leakage (test and training data too similar), the non-intersecting shell sets are balanced by number of bounding boxes to form an approximate 85% training and 15% validation split. This amounts to ≈11,745 defect bounding box instances for training and ≈4,725 defect bounding box instances for validation.

The relevant evaluation metrics for defect detection tasks are intersection over union (IoU), precision, recall, and mean average precision (mAP). IoU is a metric measuring the spatial overlap between predicted and ground-truth bounding boxes, and is formally defined as the area of the overlap between two bounding boxes (Intersection, $A \cap B$) divided by the total area encompassed by both bounding boxes (Union, $A \cup B$). This metric is typically used for detection evaluation (i.e. threshold of IoU > 0.5 means defect was detected). For the purposes of evaluation and merging defect instances, an IoU > 0.5 between two bounding boxes is considered a match.

Precision is defined as the number of true positive predictions over the total number of predictions. A high precision corresponds to low false positive predictions. Recall is the number of true positive predictions over the total number of all defects that exists. A high recall indicates a low false negative rate (i.e. large portion of defects detected). While precision and recall give insight into different aspects of a model's performance, the mAP is a metric that combines both precision and recall across various levels of confidence thresholds, giving a more comprehensive insight into a models performance. Average precision (AP) is defined as the area under the Precision-Recall curve for a given object class, and mAP is the average AP across all classes. Table II shows a comparison between the top performing models for each ResNet trained during the hyperparameter tuning process. The highest performing model after hyperparameter tuning was ResNet101, achieving a mAP of 0.97, recall of 0.92, and precision of 0.97.

TABLE II
MODEL COMPARISON: BEST PERFORMING RESNET VARIANTS

Backbone	mAP	Precision	Recall
ResNet18	0.95	0.94	0.89
ResNet34	0.96	0.97	0.81
ResNet50	0.93	0.97	0.86
ResNet101	**0.97**	**0.97**	**0.92**
ResNet152	0.96	0.97	0.90

F. Post-detection Analysis

Quantifiable selection criteria for capsule quality also requires defect volume estimation, type identification, and spatial depth estimation relative to the shell wall. To address these challenges, we implement multiplane analysis for defect volume estimation, a defect classifier, and a shell wall segmenter based on computer vision techniques.

1) Multiplane Analysis for Defect Volume Estimation: One of the limitations to our current method is that RetinaNet was designed for 2D image object detection, yet our data represents a volumetric space. The 3D tomographic data is sliced into individual images so we could apply the neural network architecture, but slicing the tomography data into stacks of images removes spatial information. Moreover, the plane orientation used for slicing images could have an impact on feature extraction. While spherical defects have symmetrical cross-sections, irregularly shaped defects have a preferential orientation for displaying a larger areal cross-section when viewed from a specific angle. If the preferential plane is not chosen, a smaller surface area will decrease the detection chances.

To account for this, we create three orthogonal data sets in the XY, XZ, and YZ planes. After inference occurs on all three planes, the volumetric information is overlaid. A density-based spatial cluster of applications with noise (DBSCAN) algorithm [11] was implemented to combine the defect information from inference on the three planes. Information about the planes the defects were detected is retained. To provide an approximate volumetric calculation of defects, the multiplane clustering results are used to merge all defects into a box of length, width, and height, based on the detections in the three orthogonal planes. An upper bound on the defect volume is calculated by taking the predicted bounding box width and height from all plane orientations. The defect volumes are calculated in voxels and converted into μm^3 using the known voxel-to-micrometer conversion factor.

While our future efforts may focus on voxel-based methods, the decision to use 2D slices with clustering was driven by

the advantage of leveraging widely available pre-trained 2D object detection models, which are more abundant and provide greater computational efficiency than their 3D counterparts. Additionally, 3D object detection methods typically require larger datasets to account for the increased dimensionality and complexity of volumetric data, which would have added significant overhead to our current approach.

2) Defect Classifier: Voids and inclusions in the shell material are known to cause unwanted perturbations and Raleigh Taylor instabilities in the implosion of the capsule [12], but no completed correlation study has been made to determine the difference in severity between different defect types. Labeling the defect type is further proven difficult due to the continued scientific understanding of defect formation, which leads to more complex defect classifications being discovered. Most recently, the defect classes have expanded from the binary lower or higher density regions to include two additional classes; petal defects in the doped region and material on the inner surface of the capsule unremoved from the mandrel leaching process [13].

Due to the on-going efforts to understand defect types in tomography data, our current model is only capable of single class object detection. To classify defects in a simple data-less way, a localized pixel intensity for all defects was compared to the surrounding area to determine a generalized higher or lower density region; classified as a void for localized darker pixel intensity or an inclusion for localized brighter pixel intensity.

3) Shell Segmentation: To date, average shells consist of a thin inner undoped HDC layer, a tungsten-doped middle layer, and a thick outer undoped HDC layer [14]. The final analysis completed on the shell is calculating the start and end locations for these 3 wall layers. To find the location of these wall layers, the pixel intensity is measured radially at one degree increments from the center of a series of tomography slices

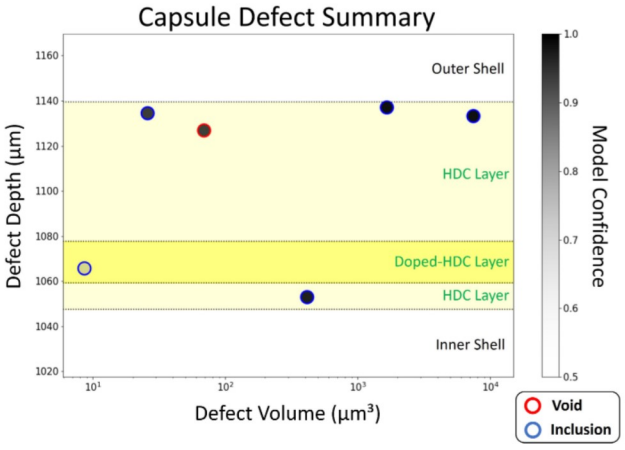

Fig. 3. All detected defects for a single shell are plotted by size and depth relative to the center of the capsule. Each circle representing a defect shows the defect type and model confidence by its perimeter color and opacity, respectively. The Y axis is clipped to show defect locations relative to the shaded shell layers (outer shell, outer HDC wall, inner doped layer, inner HDC wall, and inner shell).

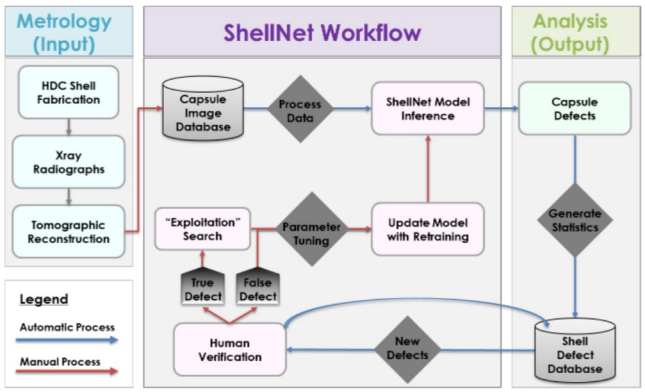

Fig. 4. Flow diagram of the semi-automated production workflow. Red arrows correspond to manual processes requiring human labor while blue arrows correspond to automated processes. The light blue metrology section shows the data creation, the purple ShellNet workflow section shows the data labeling and model training methods, and the green analysis section shows all externally reported data and statistics (includes both automated and validated data).

used to inspect the shell wall. By taking the derivative of the line intensities, we are able to locate the four highest changes in pixel intensity, which correspond to the shell wall material interfaces. A best fit circle is then used for each image to locate the center of the shell wall and radius for that image slice. After this has been completed for several images, a sinusoidal curve fit is taken for the radius distribution to calculate the center slice and shell wall thicknesses. This technique was determined to be accurate to +/- 2 voxels.

After the shell wall locations are segmented, a summary report for a single shell can be created from the culmination of all analyses from section II F, shown in Figure 3. Observed from the graph are total defects detected, the neural network detection confidence, the defect type, the defect location in the shell wall, and the defect volume. These graphs provide a metric for scientific staff to assess the general capsule quality with respect to the defects from the tomography data.

Tomographic imaging is created daily to characterize newly fabricated HDC shells for ICF experiments. In order to automate the defect detection process, train up-to-date machine-learning models and validate their performance on both historical and present data, a preliminary closed loop system has been deployed for continuous integration (CI) and delivery (CD). The generalized workflow can be seen in Figure 4.

The production workflow for x-ray analysis begins with a scientist generating the necessary radiographs used for tomographic reconstruction. These new unexamined HDC shell images synced to a version controlled image database trigger the latest version of ShellNet to run inference on a designated GPU cluster. The tomography images are pre-processed and tiled before being fed to the neural network as described in section 2A.

The output of the object detector is written to a version controlled analysis database containing predicted defect bounding boxes, tomography slice information, and confidence levels shown in green in Figure 4. Sets of adjacent instance bounding box predictions are grouped into defects by IoU to generate

This article has been accepted for publication in IEEE Transactions on Plasma Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPS.2024.3505157

the statistics mentioned in section II F. A physicist utilizes this data to either filter shells by quality for experiments or explore general trends in production quality.

After the output statistics and data are generated from ShellNet, scientists verify the accuracy of model predictions. Previously verified defects are automatically filtered, while new defects are manually verified using the "exploitation search" described in section II B. Once human verification is complete, new ground-truth data is available and regenerated statistics are added to the analysis database. When a significant amount of ground-truth data is added to the version controlled training dataset, we manually trigger the hyperparameter optimization and training cycle. At this stage, We opt not to continue training from any previous ShellNet model weights to avoid over-fitting or biasing the model toward patterns specific to previous data. New candidate object detection models are automatically proposed based on their relative performance on a validation set. The last step in the production flow diagram is to manually verify whether a top performing object detection model outperforms the current production model. Inference runs over the latest version of the image database when the latest production model is updated, completing one iteration through the CI and CD workflow.

III. Production Performance

A complete suite of algorithms was developed for implementing an object detector to analyze tomographic data, which performed well on validation data. However, while offline validation metrics are typically good indicators of performance, it does not guarantee identical performance in a real-world practical setting. Therefore, we seek to compare our analysis model to a preexisting automated method and a human expert. This comparison serves two critical purposes: firstly, to establish a robust baseline for our novel approach, and secondly, to assess its potential as a replacement for current production-level automated defect detection systems. To determine the reliability of our object detector for a production environment, we compare our neural network performance to an existing computer vision model and a human expert on previously set aside data.

A. ShellNet Performance Relative to Computer Vision

The initial comparison was made against the classical computer vision (CCV) model originally developed at GA [3]. This method detects defects by using a tolerance of intensity variations based on local pixel intensity gradients for all pixels in an image. Regions meeting a certain criteria are classified as defects. For a set of random hold out shells, we compare the total verified defects of the classical computer vision model with that detected by our best-trained object detection model. The results are displayed in Table III, with "total verified defects" referring to the total number of unique defects found by either CCV or ShellNet and subsequently verified by a human. If both methods discover the same defect, it is counted only once. For this comparison, ShellNet found 98.5% of the total verified defects with mean a precision of 94.5%, while the CCV model found 5.0% of the total verified defects with a mean precision of 40.0%. Additionally, ShellNet found 68.6% of the total defects detected by the CCV model.

TABLE III
Total Defect Detections

Shell	Total Verified Defects	CCV	ShellNet
Shell 1	36	7	36
Shell 2	143	16	135
Shell 3	64	6	63
Shell 4	246	3	246
Shell 5	91	0	91
Shell 6	114	3	112

After analyzing the results, the CCV model proved to be unstable in the phase contrast regions, lacked detection for irregular shaped defects, and was dependent on standardized image brightness, contrast, gain, and x-ray equipment. While CCV methods can mitigate these issues by manually tuning a set of parameters for each scenario, our machine-learning model has the advantage of adapting to these issues given more defect examples and data augmentation. Additionally, it was discovered that our model could find petal defects, even though they were not originally in the training dataset. This implies the machine-learning model has the additional advantage of detecting defect types not previously seen, while CCV methods require experts to explicitly describe known defect types based on mathematical models.

The results confirm that our object detector can find a magnitude more defects than the CCV method, while simultaneously detecting fewer false positive. Given that our ML model outperforms the CCV model and detects the majority of CCV detections, we can confidently replace the pre-existing CCV model for the purposes of automation.

B. Performance of ShellNet relative to Human Expert

For the purposes of comparison, a human expert is defined as a skilled scientific staff viewing tomographic images to

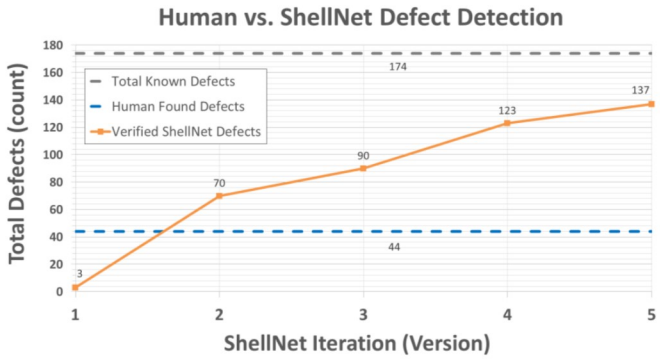

Fig. 5. Improvement in the performance of ShellNet as new data is added to the training set. Each ShellNet version shown is a completely retrained network after one complete iteration through the semi-automated workflow shown in Figure 4. The total known defects are a combination of all of the unique defects found by a human and all ShellNet iterations. Verifed ShellNet defects are expected to converge with the total, as ShellNet continues to improve with more labeled training data.

This article has been accepted for publication in IEEE Transactions on Plasma Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPS.2024.3505157

find defects in a capsule using the "needle in a haystack" approach. Humans are expected to have the most reliable defect detection precision, because they possess remarkable cognitive abilities to discern intricate visual patterns and anomalies, rarely producing false positives. However, humans are capable of missing defects and are prone to bias such as mental fatigue, training variation, expertise differences, and eye strain. Additionally, the process is time consuming and expensive for high quality data sets. On average, scientific staff take at least 3 hours to review a shell, which leaves 5.4 seconds to look at an individual image comprised of about 4 million pixels.

A single unique hold out shell was reserved to rigorously test the model's performance, encompassing 767 annotated defect bounding boxes. To prevent data leakage between the datasets used for training and production, this distinct hold-out shell is exclusively for production testing purposes only. The comparison results between the total number of defects correctly detected by ShellNet and our human expert is shown in Figure 5, plotted with respect to neural network version improvements over time. Each "version iteration" is a pass through our production workflow shown in Figure 4, after training a completely new model with the latest available training data. Human defects found is represented as a flat line because it is one single "needle in a haystack" search. Although we do not have the resources to repeat this for a single shell, we do not expect this number to fluctuate between repeat searches.

The first release of the neural network detected a limited number of defects on this production shell, initially suggesting that human-labeled data was superior. However, as the model's performance continued to improve as more data was available, it was discovered that the model could predict defects not originally caught by human experts while reviewing image slices. One scenario encountered is humans overlooking small and faint defects near larger detected defects, as exemplified in Figure 6. As data quality and training process improves, the total defects detected from the model quickly outpaces the human model. With an increase in the amount of human generated data, there should be a stronger convergence for ShellNet to detect all defects that the human is able to find in a single shell.

The final production neural network had a precision of 82.5% on the production shell, while the human model had a precision of 100.0%. The results for precision were as anticipated, while the human had better precision, the model still performed satisfactory. Conversely, ShellNet had a recall of 78.7%, while the human model had a recall of 25.3%. The network was able to find 63.6% of the defects picked up by the human.

Our test results provide statistical evidence that our model was capable of outperforming human recall in a production environment. In practice, we expect our model will detect a significant amount of defects that the human model will not identify, while also finding most defects that the human model would have already predicted.

Given the previous trend shown in Figure 5, we expect an increase in model performance over time as we continuously expand data for training. The model should eventually hit an asymptotic defect detection level, but this trend has not yet been observed from testing our latest training iterations. There is still room for improvement considering the limited amount of verified data.

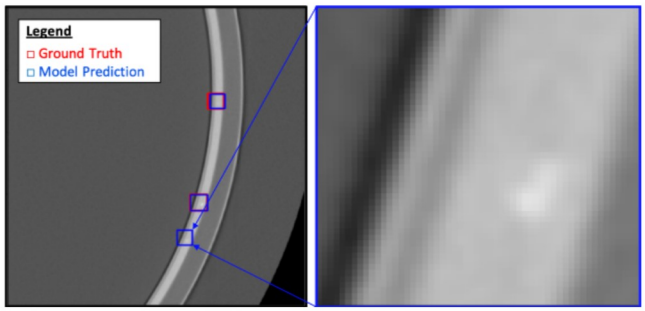

Fig. 6. The labeled ground-truth defect data is shown as red bounding boxes, while model defect predictions are displayed as blue bounding boxes. The overlapping bounding boxes show agreement between ground-truth bounding boxes and neural network predictions. The zoomed in bounding box image displayed on the right shows a "false positive" prediction, later verified as a true positive.

IV. Summary

This work has demonstrated the successful utilization of a machine-learning model for defect detection within a production level HDC shell quality assessment system. RetinaNet, a single-stage neural network object detector was trained to very accurately predict HDC shell defects, achieving over 0.97 mAP, 0.92 recall and 0.97 precision on a validation test set. The latest production model was able to outperform the recall of a human, able to detect 78.7% of all verified defects on a holdout production shell, in comparison to a human expert able to find 25.3% of total existing defects. In the future, we plan to expand the dataset size and increase the number of defect classes using active learning, as well as replacing the object detection model with 3D segmentation models for more precise volumetric analysis such as 3D UNet or YOLO2U-NET.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

REFERENCES

[1] A. L. Kritcher et al. "Design of an inertial fusion experiment exceeding the Lawson criterion for ignition" *In Physical Review E*, Vol. 106, Issue 2, American Physical Society, 2022, doi:10.1103/physreve.106.025201

[2] "National Ignition Facility surpasses long-awaited fusion milestone. (2022)" *In Physics Today*, Vol. 2022, Issue 2, p. 1213a, AIP Publishing, 2022, doi:10.1063/pt.6.2.20221213a

[3] A. Allen, C. Kong, K. Sequoia, N. G. Rice, B. Russ, M. Ratledge, and L. Lee, "Automated X-Ray Tomographic Defect Analysis in High Density Carbon Capsules," *Fusion Science and Technology*, Vol. 79, Issue 7, pp. 879–883, Informa UK Limited, 2023, doi:10.1080/15361055.2023.2169525

[4] M. Quinn, D. Orozco, K. Boehm, B. Sammuli, and W. Sweet, "A Complete System for the Autonomous Evaluation of Poly alpha-methylstyrene Mandrels", *In Fusion Science and Technology*, Vol. 79, Issue 7, pp. 791–800, Informa UK Limited, 2023, doi:10.1080/15361055.2023.2204201

[5] Amorin C, Kegelmeyer LM, Kegelmeyer WP. "A hybrid deep learning architecture for classification of microscopic damage on National Ignition Facility laser optics", *Stat Anal Data Min: The ASA Data Sci Journal*. 2019; 12: 505–513., doi:10.1002/sam.11437

[6] T. Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, 1 Feb. 2020, doi:10.1109/TPAMI.2018.2858826

[7] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, 2016, doi:10.1109/CVPR.2016.90

[8] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp. 580-587, 2014, doi:10.1109/CVPR.2014.81

[9] F. Ö. Ünel, B. O. Özkalayci and C. Çiğla, "The Power of Tiling for Small Object Detection," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, pp. 582-591, 2019, doi:10.1109/CVPRW.2019.00084

[10] P. Balaprakash, M. Salim, T. D. Uram, V. Vishwanath and S. M. Wild, "DeepHyper: Asynchronous Hyperparameter Search for Deep Neural Networks," *2018 IEEE 25th International Conference on High Performance Computing HiPC*, Bengaluru, India, pp. 42-51, 2018, doi:10.1109/HiPC.2018.00014

[11] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise" *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining KDD '96*, AAAI Press, 226–231, 1996, doi:10.1109/ICSMC.2006.384571

[12] A. B. Zylstra, et al. "Hot-spot mix in large-scale HDC implosions at NIF", *In Physics of Plasmas*, Vol. 27, Issue 9, AIP Publishing, 2020, doi:10.1063/5.0003779

[13] *High-Laser-Energy Shot Puts NIF Back on Track Toward Ignition*, National Ignition Facility and Photon Science, 7 Nov. 2022. https://lasers.llnl.gov/news/high-energy-shot-puts-nif-back-on-track-toward-ignition

[14] A. J. MacKinnon, et al. *"High-density carbon ablator experiments on the National Ignition Facility"*, 20 May 2014, In Physics of Plasmas Vol. 21, Issue 5. AIP Publishing, doi:10.1063/1.4876611