Defect Detection and Localization using 2D Slicing Method on 3D X-ray Microscopy

Aye Phyu Phyu Aung [a], **Zhuoyi Lin** [a], **Ramanpreet Singh Pahwa** [a], **Riko I Made** [b], **Senthilnath Jayavelu** [a]

[a] *Institute for Infocomm Research (I^{2}R), A*STAR, Singapore* aye_phyu_phyu_aung,lin_zhuoyi,ramanpreet_pahwa, j_senthilnath@i2r.a-star.edu.sg

[b] *Institute of Materials Research and Engineering (IMRE), A*STAR, Singapore* riko@imre.a-star.edu.sg

Presenting author

1. Introduction

Defects in semiconductor devices remain a persistent challenge. Certain defects can significantly impair device performance, ultimately affecting functionality. Their rarity often complicates detection, making non-destructive methods essential. These techniques preserve the defect's context, providing valuable insights into its root cause and guiding appropriate corrective actions. However, analyzing data from non-destructive methods, such as 3D X-ray microscopy (XRM), poses difficulties due to large data sizes and limited availability. To overcome these challenges, we propose a two-step machine learning model that performs both defect detection of the sample as either defective or non-defective, and defect localization, pinpointing the failure region within the sample.

2. Related work

In this section, we discuss 3D defect detection and object localization. Previous studies such as [1] identify anomalies by analyzing patterns in the feature space and [2] address unlabeled defect data through semi-supervised learning, relying on a small set of labeled 3D defect samples while generalizing from a larger set of unlabeled data. However, these approaches assume that defects are always present.

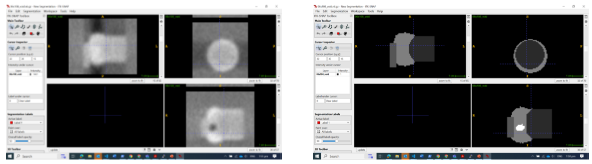
For defect localization, methods like Faster R-CNN [3] and YOLO [4] are commonly used, but they introduce challenges in our proposed pipeline of defect detection and localization. Specifically, the imbalance between defective and non-defective samples can affect model performance, and non-defective samples, which lack any defect to detect, can lead to false positives. These limitations highlight the need for a more specialized approach to 3D defect detection and localization.

3. Methodology and Implementation

3.1 Dataset and Visualization

For our dataset, we obtain 3D scans from 3D X-ray microscopy (XRM) scanner provided by Zeiss [2]. Each sample is rotated from -3°to 183°. From each 3D file, we extract unlabeled 2D slices and save as our training samples. Fig 1 compares the 3D Visualization on ITK-SNAP[1] and the results from our implementation for raw scans and ground-truth.

[1]https://sourceforge.net/projects/itk-snap/



(a) Raw Sample (b) Ground Truth

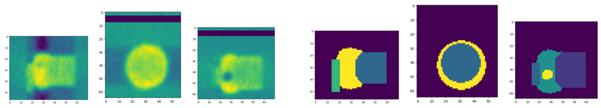
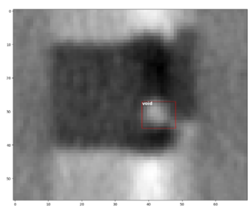


Fig. 1: ITK-SNAP (top) and our visualization (bottom) for 2D slices of each 3D-XRM Sample

3.2 Defect annotation using ground truth

We focus on voids, a common semiconductor packaging defect, i.e, trapped air pockets or gaps within materials that can degrade thermal, electrical, and mechanical performance of the device. We label our training samples by masking the ground-truth over raw scans and extract the region from the masks labeled as void. Fig 2 shows the annotated raw image and metadata with defect label as void and a bounding box around the defect. The rest of the sample are labeled as healthy. Hence, we create our dataset with 543 healthy (H) samples and 154 defect (D) samples with 80%-20% train and test data splits.



xmin	ymin	xmax	ymax	class	file	width	height
34	14	40	22	1	xzdict_216	6	8

Fig. 2: Bounding box for defects and metadata

3.3 2-step Machine Learning Model

For the detection and localization, we first try to use the commonly used object localization algorithms such as Faster R-CNN for more accuracy and YOLO-v4 for faster speed. However, the algorithms do not detect anything due to healthy sample heavy imbalance data. Hence, we first oversample the defect samples and retrain Faster R-CNN. We can detect the void defect but we found the false positive detections on healthy samples as shown in Fig 3a.

Hence, we propose a 2-step ML model for defect detection and localization by having supervised classification (Random Forest) to differentiate healthy and defect samples first. If a sample is classified as defective, we further localize the defect area by the object detection algorithm (Faster R-CNN).

In train stage, we train random forest (RF) and Faster R-CNN separately. The data splits and defect sample counts for each model are shown in Table 1. For example, we train RF with 557 samples that include 123 defects. During test stage, we first run RF to classify healthy/defect and if defective, we pass the sample to Faster R-CNN model for localization.

Table 1: Data split settings to train each model

	Train	Test	Total
RF	557 (123)	140 (31)	697 (154)
Faster R-CNN	(103)	(20)	(123)
2-step detection	-	140 (31)	-

*The defective sample counts are shown in the brackets.

4. Results

We present the qualitative detection results on test samples as well as the quantitative metrics. We performed the detection/localization experiments on 4 methods: (1) **YOLO-v4**; (2) **Faster R-CNN**; (3) **OR-CNN** i.e, Faster R-CNN trained with oversampled defective samples (434 healthy, 434 defect) and; (4) **Ours** where we train RF followed by Faster R-CNN. All methods are tested on 140 test samples.

4.1 *Qualitative Evaluation*

The first two baselines, YOLO-v4 and Faster R-CNN both fail to detect and locate the void. Fig 3a shows the false positive detection of OR-CNN method. In Fig 3b, our 2-step method correctly classifies as healthy sample and did not do localization while localizes the defective area in Fig 3c. Moreover, we also show the ground-truth target and our predicted localization comparison of our method in Fig 3d and 3e. From the results, we can see that our method outperforms the existing approaches in both classifying the healthy/defect samples as well as accurately localizing the defective area.

4.2 *Quantitative Evaluation*

In Table 2, we also present the accuracy, precision and recall w.r.t detection and localization for each method with for 140 test samples. The accuracy, precision and recall are defined as:

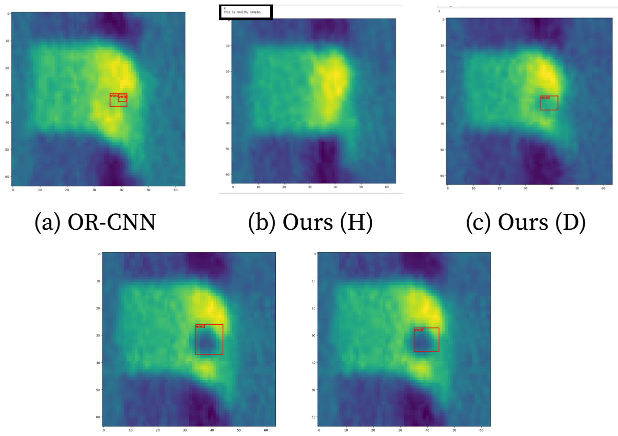
$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

5. Discussion and Conclusion

In this paper, we present a 2-step detection and localization ML model for a complete 3D-XRM failure



(a) OR-CNN (b) Ours (H) (c) Ours (D)

(d) Target (e) Prediction

Fig. 3: Results by OR-CNN and Our 2-step ML model

Table 2: Quantitative metrics over 140 test samples

	Accuracy	Precision	Recall
YOLO-v4[1]	77.86%	0%	0%
Faster R-CNN[1]	77.86%	0%	0%
OR-CNN[2]	22.14%	22.14%	100%
Ours [3]	**97.14%**	**93.54%**	**93.54%**

[1] TP=0, TN=109, FP=0, FN=31, [2] TP=31, TN=0, FP=109, FN=0
[3] TP=29, TN=107, FP=2, FN=2

analysis pipeline. While our proposed model outperforms existing methods, we still have some mispredictions. One of the reasons is due to the 2-step model being open-loop causing the heavy-reliance on RF to make the correct first-step classification. We can further improve by considering closed-loop models as future directions.

Acknowledgments

This study is supported by the Machine Learning Guided Failure Analysis & Diagnostic Capability Development for Next Generation 3D-IC Packaging at A*STAR via the IAF-PP by the Agency for Science, Technology and Research under Grant No. M23K8a0050.

References

[1] R. Pahwa, R. Chang, W. Jie, X. Xu, O.Z Min, and others. Automated detection and segmentation of HBMs in 3D X-ray images using semi-supervised deep learning. In *ECTC*, pages 1890–1897. IEEE, 2022.

[2] R. Pahwa, R. Chang, W. Jie, Z. Zhao, C. Cai, X. Xu, C.S. Foo, C.S Choong, and V.S. Rao. 3D defect detection and metrology of hbms using semi-supervised deep learning. In *2023 IEEE 73rd Electronic Components and Technology Conference (ECTC)*, pages 943–950. IEEE, 2023.

[3] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015.

[4] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma. A review of Yolo algorithm developments. *Procedia computer science*, 199:1066–1073, 2022.