ON THE USE OF COMPONENT STRUCTURAL CHARACTERISTICS FOR VOXEL SEGMENTATION IN SEMICON 3D IMAGES

Tin Lay Nwe, Ramanpreet Singh Pahwa, Richard Chang, Oo Zaw Min, Wang Jie, Yiqun Li, Dongyun Lin, Shitala Prasad, Sheng Dong

Institute for Infocomm Research (I2R), A*STAR, Singapore

ABSTRACT

Detecting defects buried inside chips is critical for failure analysis in semiconductor manufacturing. In this paper, we perform 3D voxel segmentation on 2.5D semicon chips to locate and identify defects that may be present in them. We integrate tree based Ensemble method with the Cascaded Anisotropic Convolutional Neural Networks to employ component structural characteristics of semicon 3D object in voxel segmentation process. We fabricate custom 2.5D chips purposely creating defective regions by using a specific fabrication and assembling process. Thereafter, use commercial 3D XRM tools for 3D imaging of these chips. We perform accurate 3D Object localization for each 3D x-ray scan by using a slice and fuse approach. Then, we perform voxel segmentation on logic die (integral component of semicon chip) to detect Cu-pillar, solder, and void regions (if any). The results show that we achieve state-of-the-art voxel segmentation dice scores for all three sub-components.

***Index Terms*—** Cascaded Anisotropic CNN, Ensemble Networks, 3D Deep Learning, Semicon 3D image segmentation, 3D object detection

1. INTRODUCTION

Recent advancements in Machine Learning (ML) and Deep Learning (DL) have enabled software technologies to be adapted for use with electronics manufacturing and inspection, including component design [1], optical inspection [2], and anomaly detection [3]. As the semicon components get smaller and smaller, it is getting increasingly difficult to identify defective chips which results in significant reduction in effective yield. 3D XRM tools coupled with 3D DL provide a novel solution to identify buried defects that may be present in these semicon chips.

In this paper, we will describe a methodology to fabricate sample chips which are representative of contemporary 2.5D packages, how we extracted images of subsurface features, and how Deep Learning based techniques were used for 3D Object detection and multi-class voxel segmentation. In this paper, our focus is on extracting Logic dies (or bumps) in 3D XRM scans of semicon chips and segmenting 3 regions: copper pillar (Cu-pillar), solder, and voids. A sample logic die, 3 regions for voxel segmentation and manual annotation are illustrated in Fig. 1. In Section 2 of the paper we will describe the approach used to fabricate and image the semicon chips. In Section 3, we present our methodology for 3D object detection. Proposed Ensemble network and Tree based ensemble caiCNN (Ten-caiCNN) network for multi-class voxel segmentation are discussed in Section 4. Section 5 describes data annotation and our experimental evaluation. Finally, in section 6 we present our conclusions and discuss future research directions.

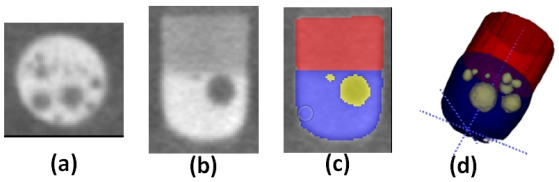

Fig. 1. Illustration of defective Logic die (or bump) in semicon chips that consist of copper pillar, solder and void regions. a), and b) display the raw 2D data slice in sagittal and coronal views respectively, c) shows the manual annotation for 2D slice in (b) and d) shows the full 3D annotation. Copper Pillar, Solder, and Voids are highlighted in red, blue, and yellow colors respectively in (c) and (d).

2. WAFER FABRICATION AND DATA ACQUISITION

We fabricated Daisy-chain silicon chips to represent the Logic dies that are stacked to form High Bandwidth Memory (HBM) cubes. The DRAM wafers were de-bonded from the temporary carrier and a wafer level underfill material was laminated over the Cu pillar bumps. The memory wafers were then diced so the chips could be stacked individually on the logic wafer, forming HBM "cubes". Thermo-compression bonding (TCB) was used to stack the memory dies on the logic wafer. We use a ZEISS Xradia 620 Versa 3D X-ray microscope (XRM) to acquire the 3D scans of semicon chips non-destructively. The imaging tool provides us with a dense voxelized scan of each chip which is approximately $1000\times1000\times1000$ resolution i.e. each scan consists of 1 billion voxels. Readers can refer to [4], [5] for more information on fabrication and data acquisition process.

3. 3D OBJECT DETECTION

3D object detection is the first step of the failure analysis process. The objective is to detect and extract the logic bumps before the segmentation and defect detection processes. The XRM tool provides high-resolution tomographies which cannot be loaded into 3D detection deep learning networks with conventional graphics cards due to the limited memory capacity. To overcome this issue, we used a slice-and-fuse approach [6], [7] that performs detection using 2D detectors and fuse the results into 3D. This approach can leverage on the accuracy and speed of 2D detection and expand the results into 3D detection results. Many 2D detectors such as SSD[8], YOLO[9] and Fast-RCNN[10], have been published in the literature and achieved high accuracy and speed. We included the YOLOv4 network as 2D detector since it has demonstrated the best detection accuracy on our data. The overall process is described as follows.

First, the 3D volume is sliced into 2D images. Then, the 2D images are loaded into the 2D detector which outputs 2D bounding boxes. Finally, we analyze all the 2D bounding boxes to estimate the 3D bounding boxes using our own algorithm based on an intersection criterion. If the intersection region between 2 bounding boxes in consecutives frames is above a certain threshold, then we combine them into a single 3D bounding box.

4. 3D SEGMENTATION

Recent works on 3D image segmentation have focused on the methods such as 3D U-Net[11], V-Net[12], CNNs with predict-refine mechanism [13], Inception modules [14], and U-Net with vibrational auto-encoder [15]. These networks have their own advantages to perform voxel segmentation for particular type of images. Recently, the study in [16] proposes Cascaded AnIsotropic Convolutional Neural Network (here, we refer the network as caiCNN) especially to perform voxel segmentation on hierarchical regions of 3D scans such as segmenting 3 hierarchical tumor regions in brain tumor MRI scans [16] [17]. The advantage of this network is the possibility to limit the focus area to detect a particular Region of Interest (ROI). As illustrated in Fig. 1, void is a subset of solder region which is a subset of logic die. Thus, we can force the network to look only inside solder region to detect void using caiCNN. We discuss the structure of logic die and our novelty in voxel segmentation in the following sub-sections.

4.1. Structure of logic component semicon object

As presented in Fig. 1, logic die consists of 3 regions: copper pillar (or Cu-pillar), solder, and void. Each region has particular characteristics. Firstly, copper pillar is made of copper and has cylindrical shape. Usually the Cu-Pillar doesn't deform during chip stacking process and hence we can make a reasonable assumption about its shape being cylindrical in the final 3D scan. Secondly, the shape of solder part of logic component can vary due to fluid dynamic nature. It can deform to any shape depending on temperature and pressure during chip packaging process. Finally, solder void (referred to as 'void') only presents inside the solder region if void exists.

4.2. Baseline Cascaded AnIsotropic network (caiCNN)

The baseline caiCNN is a cascade of 3 fully convolutional neural networks specifically designed for segmenting 3 hierarchical brain tumor regions and the background as mentioned in [16]. Given the hierarchical nature of our region of interest, especially for detecting void inside solder region, caiCNN is beneficial to us. Baseline caiCNN performs voxel segmentation of logic component into 3 different regions (or classes): copper pillar, solder, and void in 3-step cascaded process as illustrated in Fig. 2. In the first step, foreground object is segmented from the background using Whole net (W-Net). Voxels inside the detected foreground object are taken as Cu-pillar. In the second step, the Solder-Net (S-Net) detects the combined region of solder and void (if any) inside big yellow dotted line rectangle (Fig. 2) of foreground object. Detected voxels inside combined region are labeled as solder. Finally, in the third step, Void net (V-Net) detects voids (if any) inside small yellow dotted line rectangle (Fig. 2) of combined region. This cascaded process of baseline caiCNN is beneficial especially for detecting voids inside combined region. This process can avoid possible false alarms of voids in Cu-pillar region. Another advantage of caiCNN is that we can use different input patch sizes for each of the three nets: W-Net, S-Net and V-net. For example, input patch size for W-Net should be bigger than that for S-Net as W-Net is for detecting whole component over an image. Baseline caiCNN also has a disadvantage especially for detecting Cu-pillar. In the first step of caiCNN, W-Net learns the whole component (or foreground object) as the target class and all the voxels detected are primarily assigned Cu-pillar label. And, in the second step, S-Net learns combined solder and void (if any) region as the target class within whole component. There is no network specifically designed to learn well-defined cylindrical shape of Cu-pillar in the cascaded learning process of baseline caiCNN. Hence, we propose to explore Ensemble of 2-step cascaded nets to include shape information of Cu-pillar in learning process as described in the following section. Each of W-Net, S-Net and V-Net involve 10 residual blocks with anisotropic convolution, dilated convolution, and multi-scale prediction [16]. Due to page limit, please refer to WNet, TNet and ENet in [16] for detailed network architectures of W-Net, S-Net and V-Net respectively.

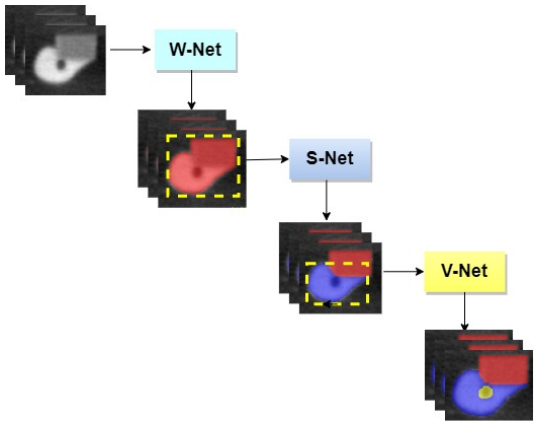

Fig. 2. Voxel segmentation of Logic component into 3 regions: Cu-pillar (in red), solder (in blue) and voids (in yellow) using 3-step cascaded process of baseline caiCNN.

4.3. Ensemble of two-step cascaded binary CNNs

Here, we propose an ensemble of two-step cascaded binary CNNs as shown in Fig. 3 to segment Cu-pillar, solder, and void regions present inside a logic die. In the first step, foreground region is segmented from the background as in baseline caiCNN. In the second step, each of the three regions is detected inside the foreground object by corresponding binary nets, C-Net, S_e-Net, and V_e-Net. The subscript, 'e', refers to ensemble method. The three binary nets: C-Net, S_e-Net and V_e-Net are trained on the same region of interest (ROI) shown using dotted yellow color rectangle of 'Cascaded-Step1' block in Fig. 3. C-Net learns Cu-pillar with cylindrical shape information as target class within whole component. And, S_e-Net and V_e-Net learn only solder and void regions respectively as target classes. All three binary nets consider the remaining voxels which are not their respective target classes inside the whole component(or foreground object) as background. The three binary nets give the three binary masks as shown in 'Cascaded-Step2' block in Fig. 3. Finally, all the binary masks are merged to produce the final multi-class segmentation mask. The Ensemble method allows the network to focus only on one type of region inside the foreground object. Hence, it has advantage on detecting Cu-pillar as C-Net can learn cylindrical shape of Cu-pillar. However, Ensemble method has disadvantage to detect

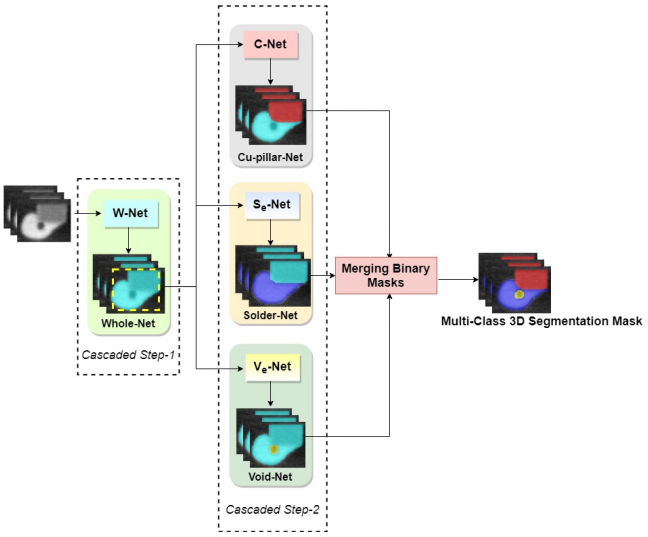

Fig. 3. Proposed Ensemble of 2-step cascaded binary CNNs for voxel segmentation of Logic component into 3 regions: Cu-pillar (in red), solder (in blue) and voids (in yellow).

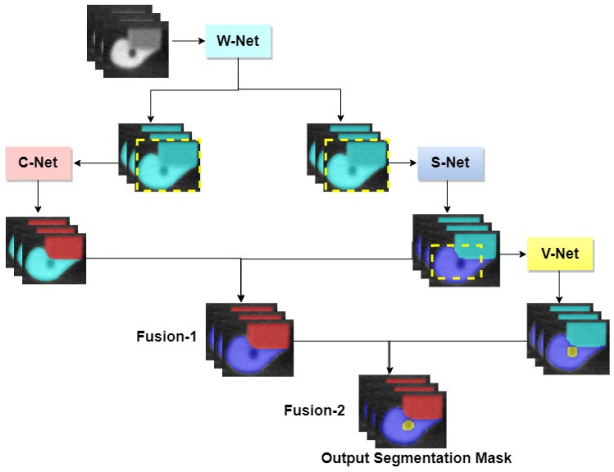

Fig. 4. Proposed Tree based ENsemble caiCNN (Ten-caiCNN) for voxel segmentation of Logic component into 3 regions: Cu-pillar (in red), solder (in blue) and voids (in yellow).

void as ROI to train the V_e-Net is on foreground object (or whole component) and this gives void false alarms in Cu-pillar region.

4.4. Tree based ensemble caiCNN (Ten-caiCNN)

As discussed earlier, baseline caiCNN has advantage in detecting voids more accurately as its 3-step cascaded process allows V-Net to focus only on combined region of solder and voids (if any). Ensemble method has advantage to segment Cu-pillar regions as the network learns structural characteristics (cylindrical shape) of Cu-pillar in segmentation process. To take advantages from both methods, we propose Tree based ENsemble caiCNN (Ten-caiCNN) as presented in Fig. 4. We add a branch for C-Net to the second step of baseline caiCNN as shown in Fig. 4. C-Net specifically learns Cu-pillar (with its cylindrical shape) inside whole component. Then, we fuse the detected masks for Cu-pillar and combined mask of solder and void as described in 'Fusion-1' of Fig. 4. We overlay the mask of Cu-pillar over combined mask to preserve the detected cylindrical shape of Cu-pillar. And, in the 'Fusion-2' step, the mask of the void detected in third step is further overlaid on top of the output mask from 'Fusion-1' to generate the final multi-class segmentation mask.

5. EXPERIMENTS AND RESULTS

5.1. 3D data annotation

We use a combination of internal tools as well as commonly available open source tool, ITK-SNAP [18] for data annotation, visualization, and quality check. Each 3D XRM scan of semicon chip contains 3 components: TSV, logic die, and memory die [5, 19, 20]. We hand annotated a subset of the components by assigning a class (TSV, Logic die, and memory die) to each 3D region and a tight 3D bounding box is drawn around individual component. Thereafter, this region is extracted to prepare dataset for 3D Bounding Box (3D-BB) detection experiments. Our dataset includes 277, 2023, and 1183 samples for TSVs, memory dies and logic dies respectively for 3D-BB detection experiments. For segmentation, we are interested to perform voxel segmentation only on logic die.We perform voxel annotations on logic die found in these extracted 3D-BBs after object detection step. We manually assign voxel labels to each of the 4 classes: Cu-pillar, solder, void (if there are voids in a sample), and background to the logic die samples. Our dataset contains 30 samples of logic die. Out of 30 logic samples, 15 logic components contained voids and the remaining 15 samples have no void. As manual voxel annotation is extremely time consuming and expensive, only a handful of logic die are annotated. Fig. 1 shows an example of the 3D annotations for logic bump consisting of Cu-pillar, solder, and voids highlighted in red, blue, and yellow colors respectively.

5.2. 3D object detection

In this subsection, we will present the results from our 3D object detection module. Our 2D detector is a Yolov4 network implemented with the darknet framework [9]. The training details for the object detector are as follows. We split the dataset into training set (85%) and testing set (15%) for all three structures (or components) separately. The pretrained model MS COCO dataset is firstly loaded and then subsequently trained on our dataset. We used 4800 epochs, with a step decay learning rate scheduling strategy consisting of initial learning rate 0.01 and subsequently multiplied with a factor 0.1. Each 3D XRM scan has a resolution of $988 \times 1015 \times 981$. They are first sliced into 981 images and the bumps are first detected in 2D. The 2D bounding boxes are processed to generate 3D bounding boxes. We evaluated our detectors using an Intersection-Over-Union (IOU) metric and mean Average Precision (mAP). We use a 0.5 threshold to determine whether the prediction is correct (True Positive) or incorrect (False Positive). The precision is then computed as Precision = $TP/(TP + FP)$. Since our detector was able to correctly detect all the 3D logic bumps, we used the IOU metric to compare the accuracy of the 3D detection. Overall, the 2D detection achieved 99.7% mAP accuracy and our 3D bounding boxes estimation achieved 96.3% IOU. The results are shown in Fig. 5.

5.3. 3D segmentation

The 3D bounding boxes of logic samples from object detection module are processed for voxel segmentation. We divide the 30 logic

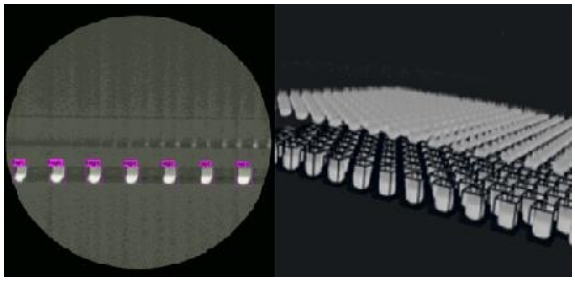

Fig. 5. 2D single image (left) and 3D results (right) for the logic bumps. The Yolov4 detector achieves 99.7% mAP accuracy and our 3D bounding boxes estimator achieves a 96.3% IOU.

Table 1. $3D$ dice score for whole component, Copper Pillar, Solder and void regions on test dataset

Method	Dice Scores			
	Whole	Cu-Pillar	Solder	Void
CaiCNN	**0.9545**	0.9050	0.9234	0.8233
Ensemble	0.9478	0.9239	0.9251	0.7961
Ten-caiCNN	0.9492	**0.9239**	**0.9257**	**0.8233**

samples into 3 disjoint subsets. A total of 8, 2 and 5 samples with voids are in train, validation and test sets respectively. And, 2 and 13 samples without void are in validation and test sets respectively. We exclude samples without void in train set as we found the resultant models to be very imprecise for void detection. Non-void data resulted in confusing our networks during the training process. For each of the networks such as W-Net mentioned in Section 4, we train a model in each of the 3 views: axial, sagittal, and coronal. Input patch sizes for W-Net, S-Net (or S_e-Net) and V-Net (or V_e-Net) are 19×128×128, 19×64×64 and 19×64×64 respectively. Input patch sizes are the same for S-Net and V-Net because voids spread all over the solder region as presented in Fig. 1(a). Each model is trained for 750 iterations on training set and the best model is selected using validation set. We compute the mean of the predictions over 3 models for the decision on assigning final voxel labels. We use the dice metric as shown below to evaluate the 3D segmentation accuracy.

$$Dice(X,Y) = \frac{2|X \cap Y| + \epsilon}{|X| + |Y| + \epsilon}$$

where X and Y are segmentation output and groundtruth masks respectively. The voxel segmentation dice scores are reported for 4 regions: whole component, Cu-pillar, solder, and voids for the test set which is unseen to the model.

5.3.1. Voxel segmentation by baseline caiCNN

First, we conduct experiments to detect 3 regions: Cu-pillar, solder and void from the background in 3-step cascaded process for our logic dataset using baseline caiCNN as discussed in Section 4.2. The 3D segmentation dice scores we achieved for the 3 regions together with whole component segmentation are listed in the 3rd row of Table 1 for baseline caiCNN. We achieve reasonably high dice scores for all the classes.

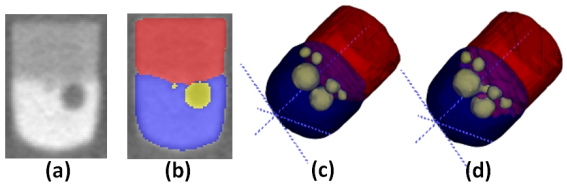

Fig. 6. We illustrate the visual segmentation results of our approach, 'Ten-caiCNN', for segmenting Cu-pillar (red color), solder (blue color), void (yellow color) and whole (combined red, blue and yellow colors). (a) displays a 2D slice of raw 3D data. (b) and (c) show our segmentation on a sample 2D virtual slice and our full 3D segmentation respectively. (d) shows the groundtruth 3D annotation. Our segmentation results are very close to the groundtruth annotation.

5.3.2. Voxel segmentation by Ensemble method

We conduct experiments by training 2-step cascaded CNNs as discussed in Section 4.3. The first step of detecting whole component is the same as in baseline caiCNN. In the second step, we detect each of the 3 regions: copper pillar, solder, and void inside the whole component individually as presented in Fig. 3. Each 2-step cascaded CNNs outputs a mask for each region inside the whole component. We fuse the masks for 3 regions to generate the voxel segmentation mask for all 3 classes. The 3D segmentation dice score for Ensemble method is listed in the 4th row of Table 1. We note that Ensemble method performs better than caiCNN to detect Cu-pillar regions. However, it doesn't perform as good as caiCNN to detect void region. This is also expected as Ensemble method involves no guidance for V_e-Net to focus only on the combined region (solder and void) where void exists. Segmentation process treats both copper pillar and solder regions with equal possibility to look for voids. In the following section, we conduct experiments using Ten-caiCNN to perform well on both copper pillar and void segmentation.

5.3.3. Tree based ENsemble caiCNN (Ten-caiCNN)

Here, we conduct experiment using Ten-caiCNN to take advantage of both caiCNN and Ensemble methods as discussed in Section 4.4. The dice scores we achieved are described in the last row of Table 1. Ten-caiCNN achieves the best dice scores for all 3 regions: Cu-pillar, solder and void. We illustrate segmentation output from Ten-caiCNN in Fig. 6. As we can see in Fig. 6(b) and 6(c), Ten-caiCNN gives very similar segmentation mask to the groundtruth annotation of Fig. 6(d).

6. CONCLUSION

In this work, we presented our approach on automated 3D segmentation and void detection for logic die of semicon chips. We described how we acquire data through fabrication and scanning processes. We presented how we extract 3D bounding boxes of logic components from large 3D XRM scans. We propose an Ensemble network and Ten-caiCNN network to perform voxel segmentation for three regions: Cu-pillar, solder and voids of logic die. The proposed Ten-caiCNN network performs better than baseline network as well as Ensemble network we explored. In future, we intend to use GAN based techniques to generate more labelled data and improve results further by improving our proposed technique using deeper networks.

7. REFERENCES

[1] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," in *Journal of Manufacturing Systems*, 2018, vol. 48, pp. 144 – 156, Special Issue on Smart Manufacturing.

[2] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," in *CIRP Annals - Manufacturing Technology*, 05 2016, vol. 65.

[3] R. S. Pahwa, J. Chao, J. Paul, Y. Li, M. T. Lay Nwe, S. Xie, A. James, A. Ambikapathi, Z. Zeng, and V. R. Chandrasekhar, "Faultnet: Faulty rail-valves detection using deep learning and computer vision," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 559–566.

[4] R. S. Pahwa, S. W. Ho, R. Qin, R. Chang, O. Z. Min, W. Jie, V. S. Rao, T. L. Nwe, Y. Yang, J. T. Neumann, R. Pichumani, and T. Gregorich, "Machine-Learning Based Methodologies for 3D X-Ray Measurement, Characterization and Optimization for Buried Structures in Advanced IC Packages," in *International Wafer Level Packaging Conference (IWLPC)*, 2020, pp. 01–07.

[5] R. S. Pahwa, T. L. Nwe, R. Chang, W. Jie, O. Z. Min, S. W. Ho, R. Qin, V. S. Rao, Y. Yang, J. T. Neumann, R. Pichumani, and T. Gregorich, "Deep Learning Analysis of 3D X-ray Images for Automated Object Detection and Attribute Measurement of Buried Package Features," in *IEEE 22nd Electronics Packaging Technology Conference (EPTC)*, 2020, pp. 221–227.

[6] A. Yang, *3D Object Detection from CT Scans using a Slice-and-fuse Approach*, Ph.D. thesis, Robotics Institute, CMU, Pittsburgh, PA, May 2019.

[7] A. Lemay, "Kidney Recognition in CT Using YOLOv3," *arXiv preprint arXiv:1910.01268*, 2019.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *European Conference on Computer Vision (ECCV)*. 2016, pp. 21–37, Springer International Publishing.

[9] A. Bochkovskiy, C-Y Wang, and H.Y. Liao, "YOLOv4: optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 04 2020.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., pp. 91–99. Curran Associates, Inc., 2015.

[11] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2016, pp. 424–432.

[12] F. Milletari, N. Navab, and S.A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," in *Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 565–571.

[13] T. L. Nwe, O. Z. Min, S. Gopalakrishnan, D. Lin, S. Prasad, S. Dong, Y. Li, and R. S. Pahwa, "Improving 3D Brain Tumor Segmentation With Predict-Refine Mechanism Using Saliency And Feature Maps," in *IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 2671–2675.

[14] D. E. Cahall, G. Rasool, N. C. Bouaynaya, and M. F. S. Hassan, "Inception modules enhance brain tumor segmentation," in *Frontiers in computational neuroscience*. 2019, vol. 13, p. 44, Frontiers.

[15] A. Myronenko, "3D MRI Brain Tumor Segmentation Using Autoencoder Regularization," in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 311–320.

[16] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, "Automatic Brain Tumor Segmentation Using Cascaded Anisotropic Convolutional Neural Networks," in *International MICCAI brainlesion workshop*. Springer, 2017, pp. 178–190.

[17] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, et al., "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," in *Transactions on medical imaging*. 2014, vol. 34, pp. 1993–2024, IEEE.

[18] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig, "User-Guided 3D Active Contour Segmentation of Anatomical Structures: Significantly Improved Efficiency and Reliability," in *Neuroimage*. 2006, vol. 31, pp. 1116–1128, Elsevier.

[19] R. S. Pahwa, T. L. Nwe, R. Chang, W. Jie, O. Z. Min, S. Gopalakrishnan, S. W. Ho, R. Qin, V. S. Rao, H. Dai, J. T. Neumann, R. Pichumani, and T. Gregorich, "Automated Attribute Measurements of Buried Package Features in 3D X-ray Images using Deep Learning," in *IEEE 71st Electronic Components and Technology Conference (ECTC)*, 2021, pp. 2196–2204.

[20] R. S. Pahwa, S. Gopalakrishnan, H. Su, O. E. Ping, H. Dai, S. W. Ho, R. Qin, and V. S. Rao, "Automated Void Detection in TSVs from 2D X-Ray Scans using Supervised Learning with 3D X-Ray Scans," in *IEEE 71st Electronic Components and Technology Conference (ECTC)*, 2021, pp. 842–849.