electronics

MDPI

Article

Synthetic Data for Semantic Segmentation: A Path to Reverse Engineering in Printed Circuit Boards

Adrian Phoulady, Hongbin Choi, Yara Suleiman, Nicholas May, Sina Shahbazmohamadi * and Pouya Tavousi *

School of Engineering, University of Connecticut, Storrs, CT 06269, USA; adrian.phoulady@uconn.edu (A.P.); hongbin.choi@uconn.edu (H.C.); yara.suleiman@uconn.edu (Y.S.); nicholas.may@uconn.edu (N.M.)
* Correspondence: sina.shahbazmohamadi@uconn.edu (S.S.); pouya.tavousi@uconn.edu (P.T.)

Abstract: This paper presents an innovative solution to the challenge of part obsolescence in microelectronics, focusing on the semantic segmentation of PCB X-ray images using deep learning. Addressing the scarcity of annotated datasets, we developed a novel method to synthesize X-ray images of PCBs, employing virtual images with predefined geometries and inherent labeling to eliminate the need for manual annotation. Our approach involves creating realistic synthetic images that mimic actual X-ray projections, enhanced by incorporating noise profiles derived from real X-ray images. Two deep learning networks, based on the U-Net architecture with a VGG-16 backbone, were trained exclusively on these synthetic datasets to segment PCB junctions and traces. The results demonstrate the effectiveness of this synthetic data-driven approach, with the networks achieving high Jaccard indices on real PCB X-ray images. This study not only offers a scalable and cost-effective alternative for dataset generation in microelectronics but also highlights the potential of synthetic data in training models for complex image analysis tasks, suggesting broad applications in various domains where data scarcity is a concern.

Keywords: deep learning; semantic segmentation; synthetic datasets; printed circuit boards; X-ray imaging; reverse engineering; automated image analysis

check for updates

Citation: Phoulady, A.; Choi, H.; Suleiman, Y.; May, N.; Shahbazmohamadi, S.; Tavousi, P. Synthetic Data for Semantic Segmentation: A Path to Reverse Engineering in Printed Circuit Boards. *Electronics* **2024**, *13*, 2353. https://doi.org/10.3390/electronics13122353

Academic Editor: Yue Wu

Received: 23 May 2024
Revised: 12 June 2024
Accepted: 12 June 2024
Published: 16 June 2024

1. Introduction

The phenomenon of part obsolescence poses a significant challenge within the microelectronics supply chain, necessitating the development of effective strategies for its mitigation. To fully express the level of criticality of this issue, consider the following example: an aircraft still in production after several decades may require a part, like a PCB, that is no longer manufactured because it is obsolete.

One strategy to address such situations is part remanufacturing, which could face serious challenges if the original design of the board is no longer available, for various reasons. In such cases, part remanufacturing can be significantly facilitated by reverse engineering techniques. In the realm of printed circuit boards (PCBs), reverse engineering can be executed using various methods, with the application of X-ray or X-ray computed tomography (CT) being particularly effective for multi-layered boards. X-ray CT provides a non-invasive means to visualize the complex internal structure of PCBs. A critical step in deciphering the design of a PCB from X-ray CT images is image segmentation. This process involves identifying and isolating different components within the PCB image to facilitate further analysis [1]. Traditionally, this segmentation has been performed manually, a method that is both time-consuming and prone to human error. Thus, automating this process is crucial and achievable through image processing and machine learning techniques, with deep learning proving to be particularly potent, often surpassing other methods and sometimes even human performance.

Pasunuri et al. explored the challenges of accurate printed circuit board (PCB) image segmentation and evaluate the suitability of several neural network techniques, including

Unet, DilatedNet, DeepLab, LinkNet, and ICNet, for extracting the bill of materials from optical images to ensure hardware assurance [2]. Li et al. proposed an automated recycling system for printed circuit boards (PCBs) with a focus on accurately segmenting surface-mounted devices (SMDs) into small devices and integrated circuits (Ics), using assembly print-based and color distribution-based methods [3]. The work presented in [4] provided a PCB semantic segmentation method using depth images and a random forest pixel classifier to locate and identify components on a PCB. Ling et al. provided a deep Siamese semantic segmentation network that combines similarity measurement with encoder–decoder architecture to detect small and irregular PCB welding defects [5]. Makwana et al. introduced PCBSegClassNet, a deep neural network designed for PCB component classification and segmentation to aid in PCB recycling, featuring a two-branch architecture that captures global context and spatial features, a texture enhancement module for precise boundaries, and a combined loss function for segmentation [6]. Liu et al. proposed Mobile-Deep, a PCB image segmentation model based on the DeepLabv3+ framework to address the challenges of inaccurate edge segmentation, segmentation holes, and slow processing speeds [7]. Qiao et al. proposed the DCNN-GC framework, combining deep convolutional neural networks and graph cut models, to segment printed circuit board wires from CT images, to overcome challenges like artifacts and complex surroundings [8].

Despite the great potential of deep learning networks for the automatic segmentation of PCBs, their application is hindered by the need for extensive, annotated datasets, which are typically rare and expensive to procure. In the context of PCB X-ray CT imaging, gathering a diverse collection of samples demands significant effort and resources, and manually labeling these images to generate reliable ground truth data requires considerable labor.

To tackle the issue of limited data availability and enhance the performance of machine learning models, data augmentation is commonly utilized. This technique involves creating new training examples by altering the existing dataset, such as by flipping, rotating, scaling, and shifting images. While effective in expanding dataset size and reducing overfitting, the scope of data augmentation is inherently limited. To bypass these constraints, data synthesis methods have been suggested and implemented in various machine learning scenarios.

Several researchers have explored the use of synthetic datasets in their studies. Wang et al. created synthetic X-ray scattering images for neural network training [9], while Wong et al. used synthetic datasets for product identification in warehouses [10]. Anantrasirichai et al. developed synthetic interferograms for volcano deformation detection from satellite imagery [11], and Kohalaka et al. applied synthetic datasets in dental implant recognition from X-ray images [12]. Unberath et al. also advocated for simulated X-ray images in training machine learning algorithms for diagnostic radiology [13]. Oesch et al. developed a "virtual data fusion" framework for automated crack detection in X-ray CT data [14]. Branikas et al. introduced a data augmentation technique using CycleGAN to improve segmentation accuracy for defect detection in critical infrastructure by generating realistic and under-represented crack images to enhance deep convolutional neural network performance [15]. Gao et al. demonstrate that training AI models on realistically simulated images from human models, a method called SyntheX, can effectively perform and even surpass real-data-trained models in X-ray image analysis, offering a scalable and ethical alternative for developing and testing AI in interventional image analysis [16]. The paper by Fridman et al. introduced ChangeChip, an unsupervised learning-based system for detecting defects in PCBs using computer vision, addressing the limitations of traditional image processing and deep learning methods by comparing images of a reference PCB with the inspected PCB, and included creation of CD-PCB, a synthetic labeled dataset for evaluating defect detection algorithms [17].

In this paper, we propose a novel data synthesis approach to address the scarcity of training data in PCB design extraction. Our method involves generating virtual images of objects with pre-defined geometries. This technique allows for the creation of an unlimited number of images without cost or effort constraints and provides inherent labeling based on known geometries, eliminating the need for manual post-processing.

There are multiple ways to generate synthetic X-ray CT image data. Physical phantoms with known geometries can be imaged using an actual X-ray CT machine [18], but this can be both time-consuming and expensive. An alternative is the use of physics-based Monte Carlo simulation algorithms like SIMIND [19] and MCNP [20] to produce realistic 2D X-ray projections [21–23], which are then reconstructed to create 3D object images. However, this approach can be computationally demanding.

Another method is ray tracing [24] for creating 2D X-ray projections of synthetic geometries, with tools like gVirtualXRay [25] facilitating this process. While less computationally intensive than Monte Carlo simulations, ray tracing does not replicate the realistic noise typically present in X-ray images. Introducing synthetic noise to the images can partially address this, but another challenge is the potential time investment required to generate numerous CAD models for the simulations.

Our research adopts an alternative approach for image synthesis, simulating the formation of X-ray images by directly drawing them, an approach that inherently lacks the real noise found in actual X-ray images due to effects like beam scattering. We counter this by introducing noise into the images. Our methods enable the scalable generation of large and diverse datasets for training deep learning networks.

Our findings demonstrate that a deep learning network, trained on synthetic data, is highly effective in performing a semantic segmentation of real X-ray CT images of PCBs. This success highlights the potential of our approach to address data scarcity issues and advance the field of design extraction in microelectronics. Moreover, our approach holds promise for application in other domains and applications.

2. Materials and Methods

Our methodology for automated semantic segmentation of PCB X-ray images comprises two primary components: (1) the generation of synthetic images with corresponding segmentation masks that closely mimic real X-ray projections, and (2) the training of a machine learning model to accurately associate these synthetic images with their respective segmentation masks.

2.1. Dataset Creation

2.1.1. Image Generation

To generate images resembling X-ray projections, we employed a straightforward technique of sketching images on a digital canvas. For simulating PCB X-ray images, we created 2D representations using basic shapes: disks and lines to mimic junctions and traces on PCBs. This involved assembling composite geometries with hollow circles of various sizes, randomly positioned to represent PCB junctions, and lines interconnecting these circles, symbolizing PCB traces. Each geometry was converted into a synthetic image by assigning intensity values, chosen randomly from a predefined range, to both the circles and lines. Figure 1 illustrates several examples of these generated images before any noise addition.

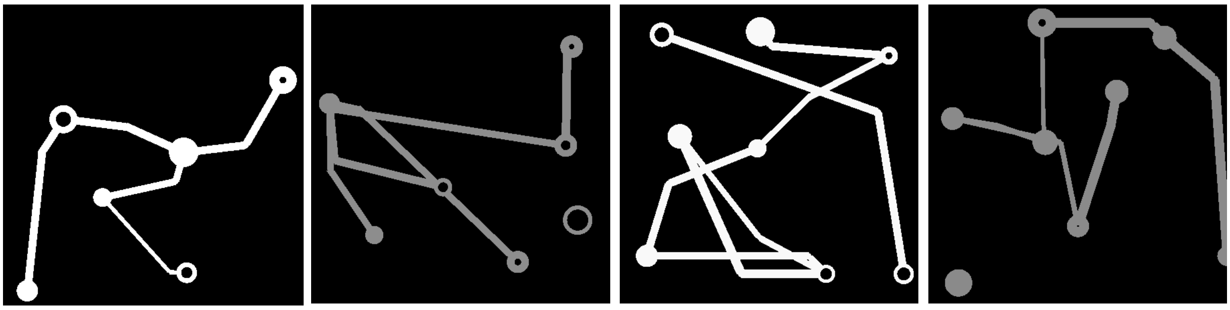

Figure 1. Examples of synthetic images generated prior to the addition of noise.

2.1.2. Noise Addition

Our method provides a rapid alternative to Monte Carlo simulations for creating X-ray-like images. However, a key limitation is the absence of real-world noise, characteristic of actual X-ray images. To address this, we introduced artificial noise to the synthesized images to enhance their realism. This was achieved by extracting a noise profile from actual X-ray images. The process involved capturing multiple consecutive air-only X-ray images under the same settings. By calculating the differences between these images and dividing the result by two, we generated various noise profiles of the imaging system. Each noise profile was larger than the size of the input images for the deep learning network, allowing us to select different subsections of noise to add to our synthetic images. The chosen noise subsection was randomly scaled with multipliers before being added to the synthetic images. Figure 2 displays two air images and their differences, which were used to generate the noise profile. The noise profile was normalized to improve contrast. While the air images exhibit a vignetting effect, the noise profile itself maintains a consistent texture.

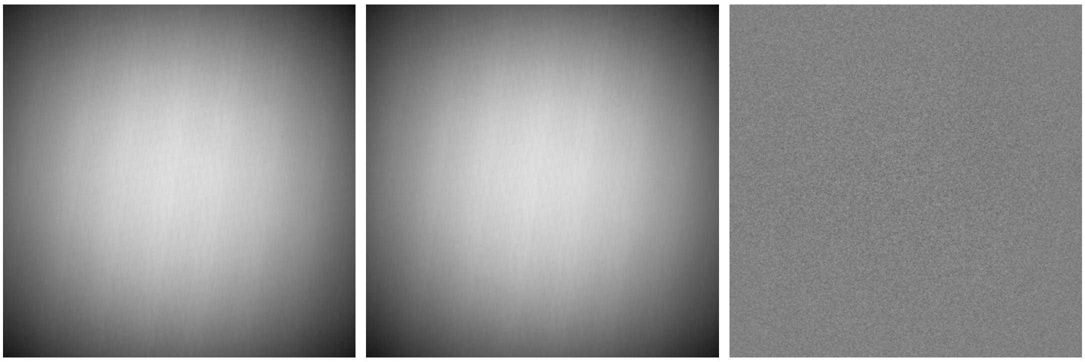

Figure 2. Air X-ray images (**left** and **middle**) used to derive the noise profile (**right**).

It should be noted that while many factors can contribute to noise, the proposed method effectively incorporates real noise—which results from a combination of several factors—into the synthetic images. This approach ensures that the effects of all contributing factors are accounted for collectively, eliminating the need to consider them separately.

An important step we took regarding noise addition was to increase the range of cases that can be covered. We implemented two main approaches. First, we doubled the range of the noise magnitude compared to what was observed in the real images. This was done to capture all possible cases, including normal imaging conditions as well as those involving poor-quality X-ray images, machine malfunctions, and inexperienced operators. In fact, we significantly stretched the noise range. Additionally, we incorporated two other noise types: blurring and contrast stretching. This further expanded the range of cases that our machine learning system can capture and cover.

2.2. Deep Learning Network Training

The copper traces and junctions together form the necessary components for retrieving the PCB design. It is critical to separate the junctions from the traces because the junctions serve as the vertices of the graph that define the PCB layout, while the traces are the edges of the graph. In this phase of our study, we directed our efforts towards the training of two deep learning networks using the synthetic dataset of PCB X-ray images we had generated. The primary objective of these networks was twofold: the first network aimed to segment the copper parts, encompassing both the junctions and the traces on the PCBs; the second network was specifically tailored to segment the junctions. By subtracting the two, the traces are then obtained. Both networks were designed based on the U-Net architecture [26,27], a choice influenced by its proven efficacy in similar image segmentation tasks and particularly due to its symmetric encoder–decoder structure, which allows for precise localization and context understanding. The encoder part of the U-Net consists of several convolutional layers with ReLU activation, followed by max-pooling layers that progressively reduce the spatial dimensions and capture high-level features. The decoder part mirrors the encoder, using transposed convolutions to upsample the feature maps and recover spatial resolution, culminating in a final output layer with a sigmoid activation function for pixel-wise classification.

To further enhance their performance, we incorporated a pre-trained VGG-16 [28,29] backbone into the network structures. VGG-16, known for its deep and simple architecture, comprises 13 convolutional layers followed by three fully connected layers. By using this pre-trained network, we leveraged its ability to extract rich feature representations from the input images, thus improving the segmentation accuracy. The weights of the VGG-16 were initialized from models pre-trained on the ImageNet dataset, providing a strong starting point for learning specific features of PCB X-ray images.

As the optimizer, we employed stochastic gradient descent with momentum, which helps accelerate convergence and escape local minima by considering the past gradients, and weight decay to prevent overfitting by penalizing large weights. We used a decaying learning rate to fine-tune the learning process, starting with a higher learning rate that gradually decreases, allowing for more precise updates to the weights as training progresses. Also, we utilized binary cross-entropy with logits loss as the loss function, which is well-suited for binary segmentation tasks and helps in effectively distinguishing the junctions and traces in the PCB X-ray images. The training of both networks was carried out using synthetic datasets.

It is important to note that the training process was conducted exclusively using synthetically generated datasets. This approach was chosen to assess the effectiveness of synthetic data in training deep learning models for real-world tasks, such as semantic segmentation in PCB X-ray imaging. For both networks, the input is a 448 × 448 gray-scale image. The output is a 448 × 448 black-and-white (BW) image that serves as the predicted label for the input image. Note that a 448 × 448 image may not be sufficient to capture the entire area of a PCB layer. For example, regarding the size of the images corresponding to one of the examples provided in the results section, we are looking at an area of approximately 13.5 mm × 13.5 mm. With a pixel size of about 14 µm, we need an image size of approximately 960 × 960 pixels to capture this area.

The flowchart below (Figure 3) summarizes the described process.

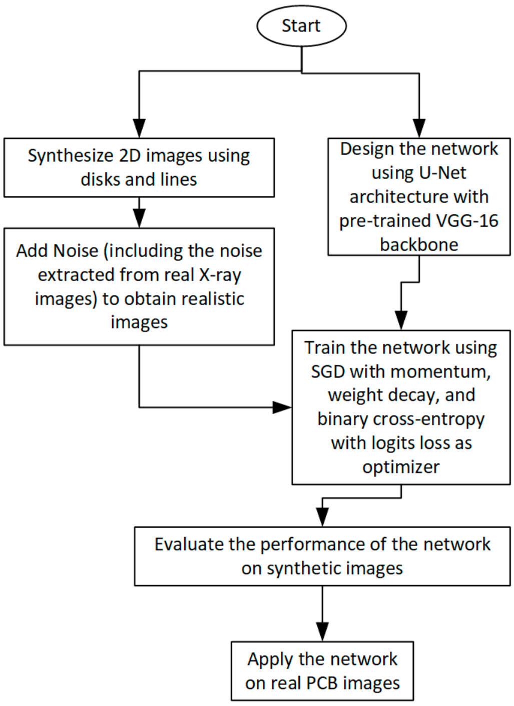
Start
Synthesize 2D images using disks and lines
Design the network using U-Net architecture with pre-trained VGG-16 backbone
Add Noise (including the noise extracted from real X-ray images) to obtain realistic images
Train the network using SGD with momentum, weight decay, and binary cross-entropy with logits loss as optimizer
Evaluate the performance of the network on synthetic images
Apply the network on real PCB images

Figure 3. The flowchart of the proposed process.

3. Results and Discussion

The research detailed in this paper sought to automate the segmentation of PCB X-ray images through deep learning techniques. Two distinct datasets were created to train separate neural networks, one for segmenting junctions and another for traces in PCB X-ray images. The process involved generating synthetic geometries on a digital canvas to simulate PCB X-ray images, followed by the addition of real noise at varying levels and patterns. More specifically, the generation of synthetic images involved multiple steps. First, a graph was generated in a random fashion by populating a corresponding connectivity matrix. The number of vertices (indicating the number of junctions) was chosen randomly from a set range of plausible values. A square matrix with the number of columns equal to the number of junctions was then created. All diagonal values of the matrix were set to zero, as there are no self-connectivities for vertices. Each off-diagonal element in the upper triangle of the matrix was randomly assigned a value of zero or one. These values were mirrored in the lower triangle to ensure symmetry. For each vertex, a random location within a specified XY range was selected on the plane, and a random diameter for that junction was chosen from a set range of values. For each edge of the graph associated with a trace on the PCB, the number of breaking points was randomly selected from a set of plausible integer values. The locations of these breaking points were then randomly determined, and the edges (traces) were created with random thicknesses, again selected from a specified range. Once the image was created, noise was added. This noise included that derived from real X-ray images by subtracting air images, as well as two additional types of noise in the form of contrast stretching and blurring. Each type of noise had a 50% chance of being applied to the image. When noise was applied, a coefficient, randomly selected from a set range of values, was multiplied by the noise profile before being applied.

Figure 4 showcases examples of these synthesized images.

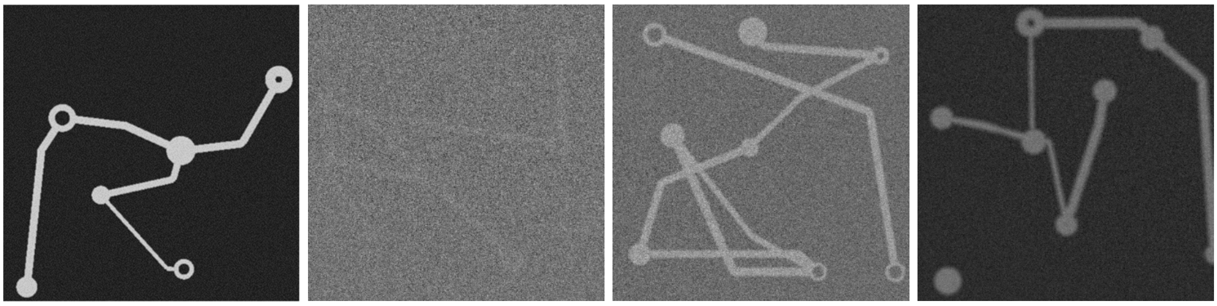

Figure 4. Synthetic images post contrast variation and addition of real noise.

The synthetic nature of these images ensured that their corresponding segmentation masks were readily available. Figure 5 presents these masks, one representing the combined traces and junctions, and the other exclusively for the junctions.

We created $\binom{5}{2} = 10$ noise profiles using five air images (1024 × 1024 resolution), From each noise profile, multiple sub-profiles of 448 × 448 pixels can be extracted. This number equals $(1024 - 448 + 1)^2$. With the 10 noise profiles, described earlier, this leads to over 3 million possible 448 × 448 noise subprofiles. By randomly selecting and scaling these subprofiles, we introduced noise to the synthetic images. We generated 20,000 synthetic images (448 × 448 pixels) with corresponding masks. These images, which did not include noise initially, each underwent a noise addition process, resulting in 20,000 noisy images. For each image, one of the over 3 million noise profiles was selected randomly. These images trained two U-net models with a pre-trained VGG-16 backbone, details of which are listed in Table 1, aimed at detecting PCB junctions and traces in X-ray images. The dataset split was 85% for training and 15% for evaluation. After 100 epochs, the board content segmentation network reached a validation loss of 0.00968, and the junction segmentation network reached 0.00580. Figure 6 presents the segmentation outputs for the synthetic images from Figure 4.

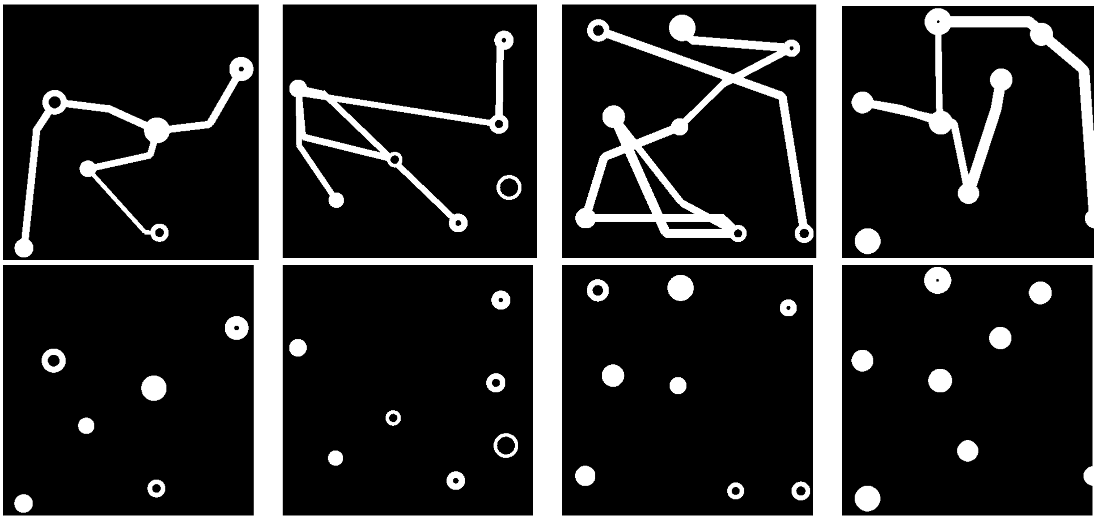

Figure 5. Board content and junction masks for synthetic images in Figure 4.

Table 1. Details of the used network structure.

Step	Description
Data Preparation	Generated synthetic PCB X-ray images with segmentation masks, including noise, blurring, and contrast variations.
Network Initialization	Initialized U-Net encoder with pre-trained VGG-16 weights from ImageNet.
Training Configuration	
- Optimizer	Used SGD with momentum (0.9) for faster convergence.
- Learning Rate	Started at 0.01 and decayed by a factor of 0.1 over every 30 epochs.
- Weight Decay	Applied 0.0005 to prevent overfitting.
- Loss Function	Binary cross-entropy with logit loss for effective binary classification.
Training Process	
- Batch Size	Set to 16 for memory efficiency and stochastic gradient descent.
- Epochs	Trained for 100 epochs with model evaluation after each epoch.
- Validation	Monitored validation loss and metrics to prevent overfitting.
Model Checkpointing	Saved best-performing models based on lowest validation loss.
Evaluation	Assessed final model using Jaccard index on real PCB X-ray images, achieving high accuracy (>0.9).

To quantitatively evaluate the networks' accuracy on real PCB X-ray images, we computed the mean Jaccard index for 20 real images. The Jaccard Index, also known as the Jaccard similarity coefficient, is a statistic used to measure the similarity and diversity of sample sets. It is calculated as the size of the intersection divided by the size of the union of two sets. For example, if we have two sets A and B, the Jaccard Index (J) can be calculated using the following formula: $J\ (A, B) = |A \cap B| / |A \cup B|$.

Here, $|A \cap B|$ represents the number of elements common to both sets, and $|A \cup B|$ represents the total number of unique elements across both sets. The result is a value between zero and one, where zero means no similarity (no shared elements) and one means complete similarity (all elements are shared).

This approach provides a clear metric for assessing how accurately the network identifies and classifies different components in the images, reflecting both the precision and recall of the model. Values closer to one indicate higher accuracies, as they show a strong match between the automated and manual segmentations.

The networks achieved Jaccard indices of 0.924 for traces and 0.937 for junctions. With the Jaccard index values obtained herein being greater than 0.9, we can conclude a high accuracy for the segmentation effort. Figure 7 illustrates a real PCB X-ray image alongside its fused segmentation output, displaying effective segmentation by both networks.

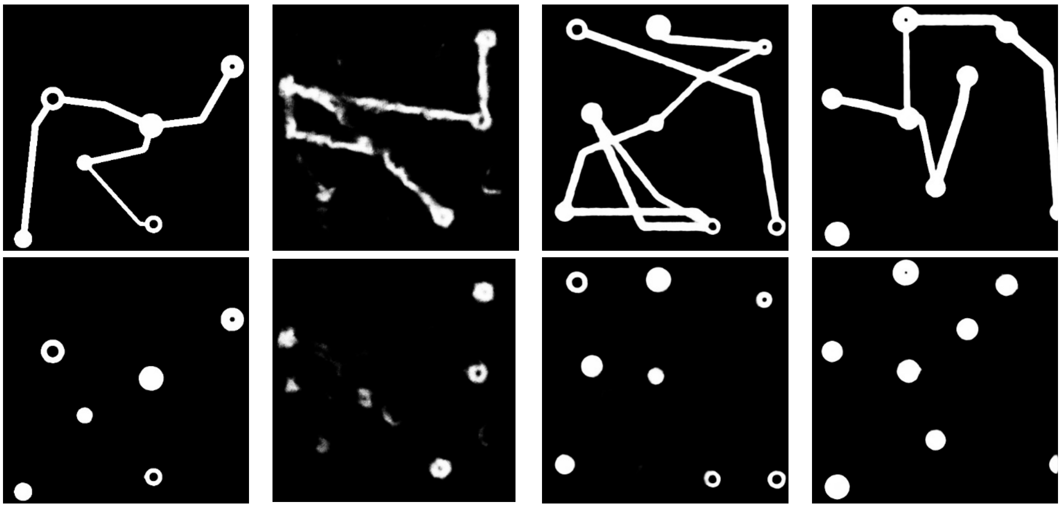

Figure 6. Results for images in Figure 4. (**Top**): segmentation of board content; (**Bottom**): segmentation of junctions.

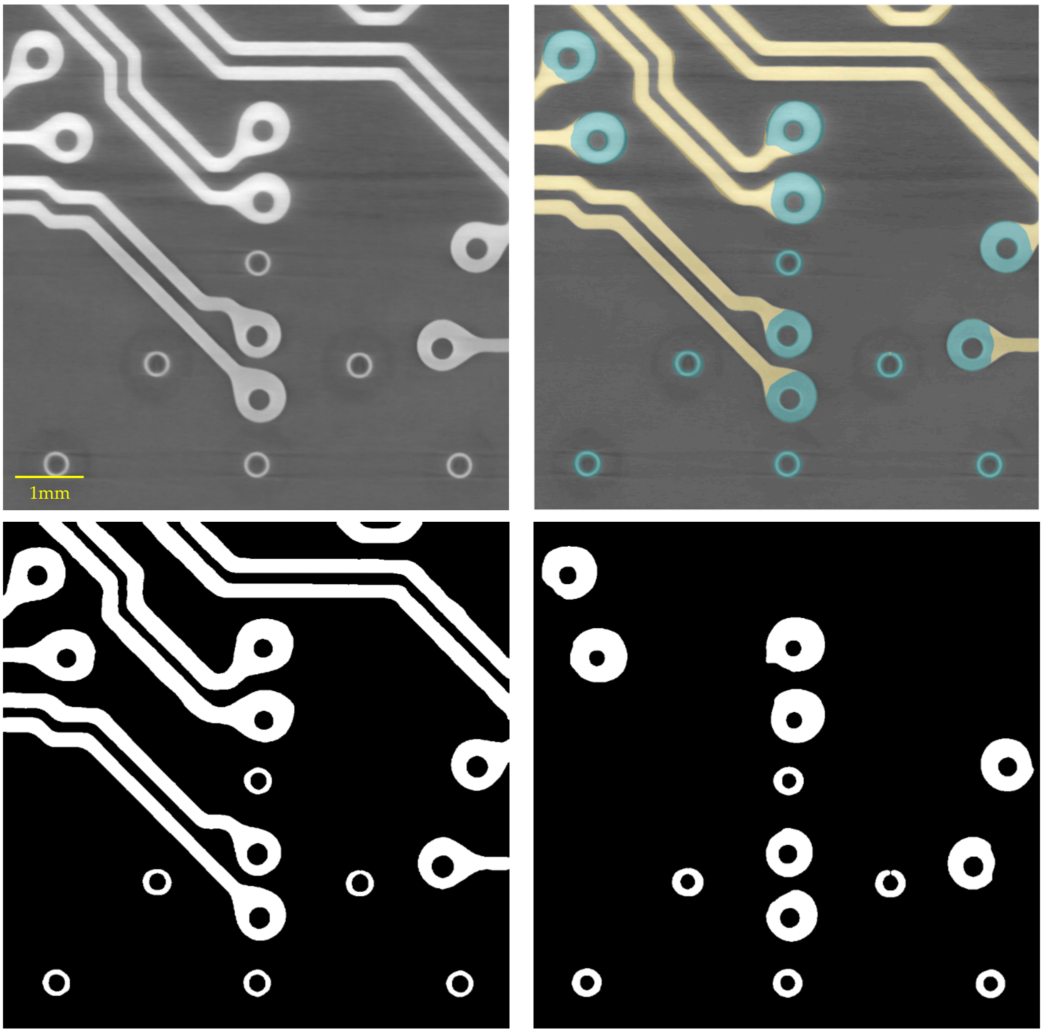

Figure 7. (**Top**): real PCB X-ray image and fused segmentation; **Bottom**): post-processed segmentation for board content and junctions.

For larger PCB X-ray images, we divided them into 448 × 448 segments with 50% overlap. After network prediction, we combined these into a normalized inference matrix.

Figure 8 shows the fused segmentation result for these larger boards, highlighting the effective segmentation by the deep-learning networks. Post-processing involved thresholding, erosion, connected component analysis, small component elimination, and dilation.

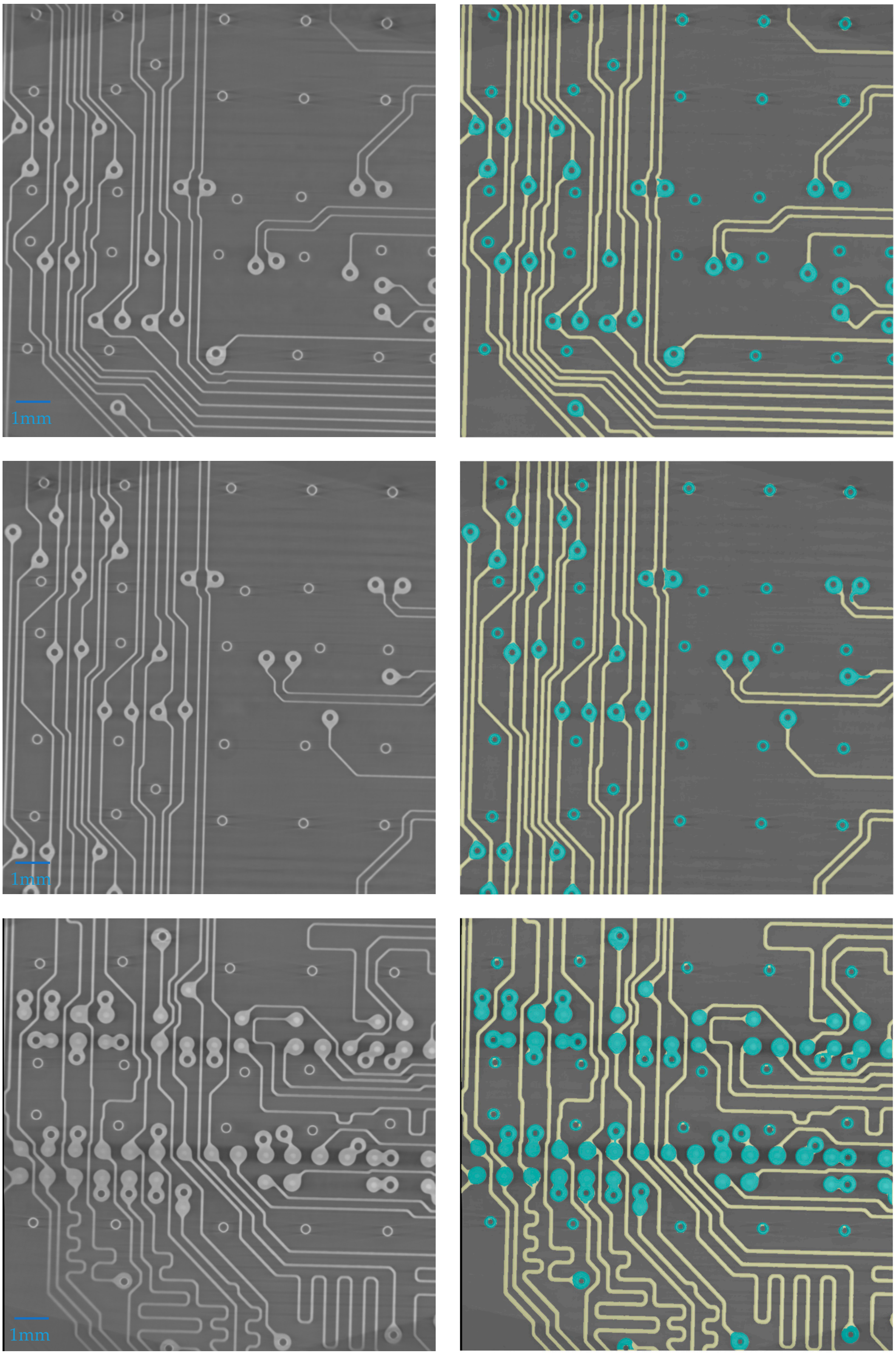

Figure 8. Fused segmentation results for larger PCBs. The left images are real X-ray images of PCB samples, and the right images depict the results of the trained network's semantic segmentation. In

the right images, the predicted labels—which are outputs from the network—are overlayed onto the original X-ray images. These fused images clearly demonstrate the network's success in precisely predicting the correct labels for each pixel, resulting in highly accurate semantic segmentation.

Our approach eliminates the need for costly image acquisition and manual labeling for training segmentation algorithms. By training on a synthetic dataset that represents various PCB variations, our solution offers a generic approach suitable for diverse scenarios and samples. Further, the proposed approach is scalable and can accommodate more complex segmentation tasks with a one-time additional computational cost for generating more diverse synthetic images. It must be noted that a key factor in the success of the proposed data synthesis method is to ensure that the synthetic data capture the range of variations observed in real-world data. This balance involves using detailed yet computationally feasible synthetic models and might require a few real images for system noise calibration. If real images are unavailable for calibration, expanding the diversity of synthetic data to cover more noise profiles may incur additional computational costs for both data synthesis and training. The synthetic data presented in Figure 4 that are created with the described approach may appear simple. However, when the network is exposed to numerous such patterns, it is successful in semantically segmenting a wide range of metal trace patterns, as showcased in the example results.

3.1. Discussion of Tradeoff between Synthetic Image Realism and Computational Complexity

Images can be synthesized with varying degrees of realism. They can either

1. Closely resemble real images through physics-based simulations of X-ray images. This can be achieved via a Monte Carlo simulation for highly realistic images or ray tracing followed by noise addition. The work presented in [30] provides an example where real and synthetic images are indistinguishable.
2. Be simpler, capturing only the essential features needed for training the machine learning algorithm.

There is a trade-off between the level of detail in the image and the computational complexity of generating synthetic images. Our findings indicate that using disks and lines with added noise strikes an effective balance, providing sufficient detail for training ML algorithms for the automatic segmentation of PCB X-ray images while remaining computationally affordable. Notably, even though the synthetic images are not intentionally highly realistic, they effectively fulfill the training requirements of the ML algorithms.

3.2. Scope of Application

The use of the proposed data synthesis approach is multifaceted. In this paper, we primarily focused on utilizing synthetic data to train machine learning algorithms for semantic segmentation aimed at the automated reverse engineering of PCBs from X-ray images. However, this approach also holds the potential for the automated detection of possible root causes of failures. The use of synthetic data for the automatic identification of defects in X-ray images by machine learning algorithms has been explored in [30], addressing data scarcity from defective parts and identification of diverse defects, such as missing bond wires in ICs, delamination effects, cracks, etc.

3.3. Comparison with Other Methods

Table 2 provides a comparison between different categories of methods that can be used for the segmentation of PCB images.

On one hand, traditional manual segmentation involves high labor costs and a slow process, as it depends on how many individuals work in parallel to perform the segmentation. This method is also subject to human error, and although it can have a wide scope of application, it is limited to cases familiar to human vision. Traditional image processing methods, such as adaptive thresholding, may separate the copper content from the rest in PCBs. However, the performance of these methods in correctly segmenting the images depends largely on the parameters of the thresholding method. Furthermore, these methods are not capable of semantic segmentation to distinguish junctions. Rule-based image processing techniques can potentially address this to some extent if rules are applied to the segmented images from the thresholding methods. These techniques may be used for detecting disks as junctions, but they are not easily generalizable to complex shapes. Machine learning offers a promising alternative. However, the conventional use of manually annotated data in machine learning-based semantic segmentation has two significant issues: high instrument cost for image acquisition and high labor cost for labeling efforts to prepare training data. However, using inherently labeled synthetic data for training addresses these issues effectively, as these methods require minimal image acquisition for calibration and zero labor for data labeling.

Table 2. Comparison of proposed method with other methods.

Method	Cost (Labor/Instrument)	Efficiency	Reliability and Effectiveness	Universality
Manual Semantic Segmentation	High labor cost	Slow process (dependent on how many individuals work in parallel to perform the segmentation)	Subject to human error	Wide scope of application, although limited to cases that are familiar to human vision
Thresholding-Based Segmentation	Low cost	Fast	Not capable of semantic segmentation (a follow-up semantic segmentation process is required)	Very case-specific. Thresholding parameters need to be adjusted per each type of image
Rule-Based Segmentation	Computational cost increases with the complexity of images or components that need to be segmented	Fast	Potential errors due to unforeseen scenarios that cannot be addressed by the set rules	Requires extremely complex algorithms to cover a wide range of possible cases
Machine-Learning-Based Semantic Segmentation Using Manually Annotated Data for Training	High instrument cost for image acquisition and high labor cost for labeling efforts to prepare training data	Fast	Highly reliable when sufficiently trained	Can cover a wide range of cases when sufficiently trained
Machine-Learning-Based Semantic Segmentation Using Inherently Labeled Synthetic Data for Training	Minimal image acquisition (for calibration) and zero labor	Fast	Highly reliable when sufficiently trained	Can cover a wide range of cases when sufficiently trained

4. Conclusions

This paper has presented a comprehensive study on the use of synthetic data for the training of deep learning networks in the context of semantic segmentation of PCB X-ray images. Our research has demonstrated that synthetic data generation is not only feasible but also highly effective in overcoming the challenges associated with limited real-world data availability in the domain of microelectronics.

We successfully developed and implemented a novel method for creating synthetic X-ray images of PCBs, incorporating noise profiles derived from real X-ray images to enhance realism. The use of this approach allowed us to bypass the limitations of traditional methods like physical phantoms or computationally intensive Monte Carlo simulations, providing a scalable and cost-effective alternative for data generation. Training two U-net models with a pre-trained VGG-16 backbone exclusively on these synthetic datasets yielded impressive results. The networks achieved high Jaccard indices in segmenting both traces and junctions of real PCB X-ray images, underscoring the potential of synthetic data to effectively train deep learning models for complex, real-world tasks.

Moreover, our approach has significant implications for the broader field of image segmentation and analysis. The scalability and adaptability of this method suggest its applicability to other domains where data scarcity is a concern. By generating diverse synthetic datasets tailored to specific requirements, researchers and practitioners can train robust models without the extensive costs and efforts typically associated with real data collection and labeling.

One limitation of the proposed approach is that it relies on the noise profile of the instrument used for sample imaging to enhance the realism of the synthetic images. In future work, we plan to replace this device-specific noise profile with a generic one.

In conclusion, our study validates the effectiveness of synthetic data in training deep learning networks for semantic segmentation tasks in microelectronics, offering a promising direction for future research and application in various fields requiring advanced image analysis.

Author Contributions: Conceptualization, A.P., S.S. and P.T.; Methodology, A.P. and P.T.; Software, A.P. and P.T.; Validation, A.P. and P.T.; Formal analysis, A.P. and P.T.; Investigation, A.P., H.C., Y.S., N.M., S.S. and P.T.; Resources, S.S. and P.T.; Writing—original draft, A.P. and P.T.; Writing—review & editing, H.C., Y.S., N.M. and P.T.; Visualization, A.P.; Supervision, S.S. and P.T.; Project administration, S.S. and P.T.; Funding acquisition, S.S. and P.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by UConn REFINE center.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: Pouya Tavousi and Sina Shahbazmohamadi declare involvement in a startup company, Aerocyonics Imaging LLC, that may engage in the commercialization of the presented research.

References

1. Asadizanjani, N.; Shahbazmohamadi, S.; Tehranipoor, M.; Forte, D. Non-destructive pcb reverse engineering using x-ray micro computed tomography. In Proceedings of the ISTFA 2015, Portland, OR, USA, 1–5 November 2015; pp. 164–172.
2. Pasunuri, A.; Jessurun, N.; Dizon-Paradis, O.P.; Asadizanjani, N. A comparison of neural networks for pcb component segmentation. In Proceedings of the 2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), Virtual, 13 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 113–123.
3. Li, W.; Esders, B.; Breier, M. SMD segmentation for automated PCB recycling. In Proceedings of the 2013 11th IEEE International Conference on Industrial Informatics (INDIN), Bochum, Germany, 29–31 July 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 65–70.
4. Li, D.; Li, C.; Chen, C.; Zhao, Z. Semantic segmentation of a printed circuit board for component recognition based on depth images. *Sensors* **2020**, *20*, 5318. [CrossRef] [PubMed]
5. Ling, Z.; Zhang, A.; Ma, D.; Shi, Y.; Wen, H. Deep Siamese semantic segmentation network for PCB welding defect detection. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5006511. [CrossRef]
6. Makwana, D.; Mittal, S. PCBSegClassNet—A light-weight network for segmentation and classification of PCB component. *Expert Syst. Appl.* **2023**, *225*, 120029. [CrossRef]
7. Liu, L.; Ke, C.; Lin, H. Mobile-Deep Based PCB Image Segmentation Algorithm Research. *Comput. Mater. Contin.* **2023**, *77*, 2443–2461. [CrossRef]
8. Qiao, K.; Zeng, L.; Chen, J.; Hai, J.; Yan, B. Wire segmentation for printed circuit board using deep convolutional neural network and graph cut model. *IET Image Process.* **2018**, *12*, 793–800. [CrossRef]
9. Wang, B.; Yager, K.; Yu, D.; Hoai, M. X-Ray Scattering Image Classification Using Deep Learning. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017.
10. Wong, M.Z.; Kunii, K.; Baylis, M.; Ong, W.H.; Kroupa, P.; Koller, S. Synthetic dataset generation for object-to-model deep learning in industrial applications. *PeerJ Comput. Sci.* **2019**, *5*, e222. [CrossRef] [PubMed]
11. Anantrasirichai, N.; Biggs, J.; Albino, F.; Bull, D. A deep learning approach to detecting volcano deformation from satellite imagery using synthetic datasets. *Remote Sens. Environ.* **2019**, *230*, 111179. [CrossRef]
12. Kohlakala, A.; Coetzer, J.; Bertels, J.; Vandermeulen, D. Deep learning-based dental implant recognition using synthetic X-ray images. *Med. Biol. Eng. Comput.* **2022**, *60*, 2951–2968. [CrossRef]

13. Unberath, M.; Zaech, J.-N.; Gao, C.; Bier, B.; Goldmann, F.; Lee, S.C.; Fotouhi, J.; Taylor, R.; Armand, M.; Navab, N. Enabling machine learning in X-ray-based procedures via realistic simulation of image formation. *Int. J. Comput. Assist. Radiol. Surg.* **2019**, *14*, 1517–1528. [CrossRef]
14. Oesch, T.; Weise, F.; Bruno, G. Detection and quantification of cracking in concrete aggregate through virtual data fusion of X-ray computed tomography images. *Materials* **2020**, *13*, 3921. [CrossRef]
15. Branikas, E.; Murray, P.; West, G. A novel data augmentation method for improved visual crack detection using generative adversarial networks. *IEEE Access* **2023**, *11*, 22051–22059. [CrossRef]
16. Gao, C.; Killeen, B.D.; Hu, Y.; Grupp, R.B.; Taylor, R.H.; Armand, M.; Unberath, M. Synthetic data accelerates the development of generalizable learning-based algorithms for X-ray image analysis. *Nat. Mach. Intell.* **2023**, *5*, 294–308. [CrossRef] [PubMed]
17. Fridman, Y.; Rusanovsky, M.; Oren, G. ChangeChip: A reference-based unsupervised change detection for PCB defect detection. In Proceedings of the 2021 IEEE Physical Assurance and Inspection of Electronics (PAINE), Washington, DC, USA, 30 November 2021–2 December 2021; pp. 1–8. [CrossRef]
18. Glick, S.J.; Ikejimba, L.C. Advances in digital and physical anthropomorphic breast phantoms for x-ray imaging. *Med. Phys.* **2018**, *45*, e870–e885. [CrossRef] [PubMed]
19. Toossi, M.B.; Islamian, J.P.; Momennezhad, M.; Ljungberg, M.; Naseri, S.H. SIMIND Monte Carlo simulation of a single photon emission CT. *J. Med. Phys./Assoc. Med. Phys. India* **2010**, *35*, 42.
20. Forster, R.A.; Cox, L.J.; Barrett, R.F.; Booth, T.E.; Briesmeister, J.F.; Brown, F.B.; Bull, J.S.; Geisler, G.C.; Goorley, J.T.; Mosteller, R.D.; et al. MCNP™ version 5. *Nucl. Instrum. Methods Phys. Res. Sect. B Beam Interact. Mater. At.* **2004**, *213*, 82–86. [CrossRef]
21. Spezi, E.; Downes, P.; Radu, E.; Jarvis, R. Monte Carlo simulation of an x-ray volume imaging cone beam CT unit. *Med. Phys.* **2009**, *36*, 127–136. [CrossRef] [PubMed]
22. Bonin, A.; Chalmond, B.; Lavayssière, B. Monte-Carlo simulation of industrial radiography images and experimental designs. *NDT E Int.* **2002**, *35*, 503–510. [CrossRef]
23. Giersch, J.; Durst, J. Monte Carlo simulations in X-ray imaging. *Nucl. Instrum. Methods Phys. Res. Sect. B Beam Interact. Mater. At.* **2008**, *591*, 300–305. [CrossRef]
24. Glassner, A.S. *An introduction to Ray Tracing*; Morgan Kaufmann: Burlington, MA, USA, 1989.
25. Sujar, A.; Meuleman, A.; Villard, P.-F.; Garcia, M.; Vidal, F.P. gVirtualXRay: Virtual x-ray imaging library on GPU. In *Computer Graphics and Visual Computing*; The Eurographics Association: Manchester, UK, 2017.
26. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, 5–9 October 2015; pp. 234–241.
27. Yin, X.X.; Sun, L.; Fu, Y.; Lu, R.; Zhang, Y. U-net-based medical image segmentation. *J. Healthc. Eng.* **2022**, *2022*, 4189781. [CrossRef]
28. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
29. Mascarenhas, S.; Agarwal, M. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In Proceedings of the 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), Bengaluru, India, 19–21 November 2021; IEEE: Piscataway, NJ, USA, 2021; Volume 1, pp. 96–99.
30. Choi, A.; Suleiman, Y.; Choi, H.; Moore, T.; May, N.; Shahbazmohamadi, S.; Tavousi, P. Synthetic data augmentation to enhance manual and automated defect detection in microelectronics. *Microelectron. Reliab.* **2023**, *150*, 115220.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.