AI-Based Microsection Measurement Framework Using ComfyUI Workflow for Flexible Printed Circuit Board

Ting-Ting Chang[1], Jo-Yu Li[2], and Chia-Yu Lin[2(✉)]

[1] Department of Information Management, National Central University, Taoyuan, Taiwan

[2] Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan

sallylin0121@ncu.edu.tw

Abstract. Microsectioning is a destructive testing method extensively employed in the flexible printed circuit board (FPC) fabrication industry to assess the structural integrity and quality of FPCs. FPCs are essential components in various electronic devices due to their flexibility, lightweight nature, and ability to fit into complex shapes and spaces. A cross-section, or microsection, involves obtaining a thin slice of the FPC to expose its internal structure. During cross-section analysis, operators manually measure the thickness of FPC components, such as copper layers, OSC layers, and FSL layers. However, this manual process can lead to inconsistencies and difficulties in establishing standardized measurement procedures. To address these challenges, we propose an "AI-based Microsection Measurement Framework using ComfyUI Workflow" for FPC. This framework comprises five key modules: the target detection module, the image preprocessing and augmentation module, the AI model building and fine-tuning module, the measurement algorithm development module, and the ComfyUI visualization module. The measurement algorithm uses predicted masks from the AI model to perform precise measurements, while the visualization module plots these results directly onto the original image for easy review. In addition, we evaluate the proposed framework on two microsection types. Our experiments demonstrate that the measurement accuracy reaches an error margin of 0 pixels. Compared to the existing method, we provide a unified, faster, and lower labor cost measurement framework.

Keywords: Microsection Measurement · printed circuit board · deep learning · ComfyUI

1 Introduction

Microscopic slicing is a critical measurement method in PCB manufacturing, playing an essential role in quality control and monitoring [9]. This technology allows the industry to inspect the internal structures of PCBs and perform

W.-T. Chu et al. (Eds.): TAAI 2024, CCIS 2415, pp. 304–318, 2025.
https://doi.org/10.1007/978-981-96-4596-1_23

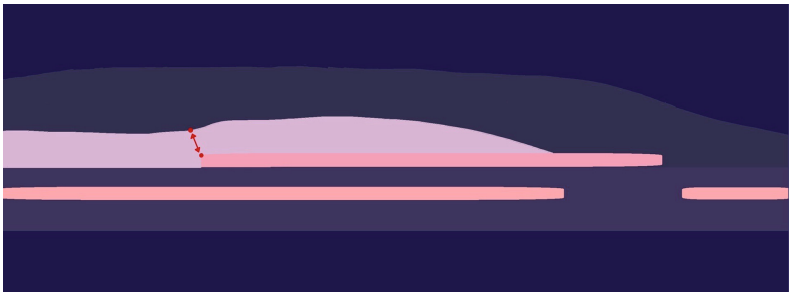

Fig. 1. A sample microsection of silver pastes. The red line is the distance that needs to be known, and the red dot is the position that needs to be predicted. (Color figure online)

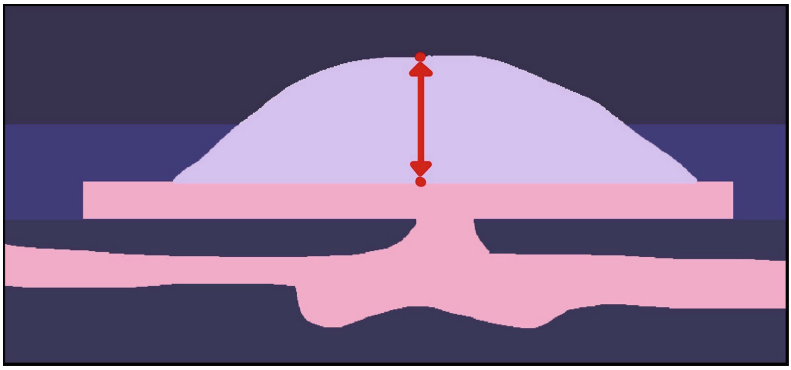

Fig. 2. A sample microsection of mental layers.

precise measurements, ensuring thorough quality control throughout the production process. It also enables accurate and efficient validation and verification of hardware boards [3,13]. Due to privacy policy restrictions, we cannot present the dataset directly. We reconstruct representative images of two types of PCB microsections. Figure 1 and Fig. 2 shows a sample of silver paste and metal layer, respectively. The dots represent the coordinates for measuring the distance. Traditionally, microscopic slice measurements in factories have been conducted manually, heavily dependent on human observation. This manual approach, while widely used, often results in inconsistencies or omissions. Given the need for micron-level accuracy in PCB measurements, manual inspections are labor-intensive, and real-time continuous monitoring is challenging [2]. These factors have led to substantial time and labor costs.

With the rise of smart factories, many production lines have adopted automation, reducing the dependence on manual labor. However, microscopic slicing presents complex measurement challenges that are difficult to fully automate. While deep learning models have shown potential in point detection and mea-

surement tasks, they often fall short of the required precision-particularly within a 2-pixel margin of error. Additionally, adapting models to different measurement tasks in varied scenarios is resource-intensive, requiring significant time and effort.

To address these challenges, our study proposes an AI-based slice measurement algorithm integrated with the ComfyUI workflow. This approach leverages the semantic segmentation of the YOLOv8 model to identify specific measurement regions and generate corresponding masks. These masks are then used to compute task-specific metrics, achieving precision beyond the 2-pixel margin requirement. The user-friendly ComfyUI interface further enhances the system's usability and efficiency in practical applications.

The remainder of this paper is structured as follows: Sect. 2 provides an in-depth analysis of related works on microsection measurement, instance segmentation, and ComfyUI interface. Section 3 outlines the data used in the study and provides a thorough explanation of the five modules of our research architecture. Section 4 discusses the evaluation metrics, experimental setup, and analysis of the results. Finally, Sect. 5 concludes the paper and explores potential future research directions.

2 Related Work

Printed circuit boards (PCBs) are essential substrates for connecting and securing electronic components, found in various devices like computers, cell phones, televisions, and cameras [6]. Flexible printed circuit boards (FPCBs) offered a lightweight and versatile alternative to rigid PCBs, enabling more compact designs, reducing size and weight, and lowering installation costs [11]. With the continuous advancements and improved functionalities of electronic products in recent years, the demands for high performance and quality in printed circuit boards (PCBs) have been increasing [18]. The inspection of FPCBs is crucial for maintaining product quality, however, it can be very expensive [20]. [12] was centered on automated optical inspection (AOI), which was increasingly preferred for its precise results and cost-effectiveness compared to manual techniques. The key to improving the accuracy of AOI lay in effective segmentation, which involved tasks such as edge detection and the extraction of the region of interest (ROI). This process helped minimize irrelevant factors and enhanced the performance of the classifier [17]. To detect PCB defects, [16] developed a non-contact, reference-based PCB defect detection and classification system utilizing a morphological image segmentation algorithm and basic image processing techniques. Preliminary findings indicated that specific PCB defects were associated with particular groups, suggesting the potential to improve detection by employing segmentation methods.

With the increasing popularity of deep learning, algorithms based on deep learning were rapidly developing and were widely used in surface defect detection [5]. Using deep learning had higher accuracy for landmark technology. In [4,14], the methods of PCB defect detection in recent years have been

reviewed. Researchers explored various uses of neural networks in image segmentation and defect classification alongside traditional computer vision methods. Object detection networks like YOLOv8 [10], R-CNN [8], Fast R-CNN [7], and Faster R-CNN [21] have been studied extensively. [22] compared the performance of the one-stage YOLOv8 and the two-stage Mask R-CNN machine learning models, for instance, segmentation across different orchard conditions. It showed that YOLOv8 exhibits superior accuracy and efficiency compared to two-stage models, particularly Mask R-CNN, suggesting its suitability for smart and automated orchard operations. Additionally, an enhanced YOLOv8-Seg model for underwater trash instance segmentation [1] demonstrated effective integration of semantic segmentation and object detection, offering precise masks and suitability for real-time applications.

To facilitate user interaction with ComfyUI, a node-based editor for Stable Diffusion (SD) that enables users to create complex workflows without the need for programming [19], an interface was developed to support VersusDebias and debiasing research [15]. This interface is adaptable to various models by simply modifying workflow files and offers a comprehensive API for seamless integration with external programs.

3 Methodology

The architecture and framework of our study are shown in Fig. 3. The research architecture consists of five modules: the target detection module, image preprocessing and augmentation module, build and finetune AI Models, measurement algorithm development Module, and the ComfyUI visualization module. In this section, we will discuss each of these modules separately in detail.

3.1 Target Detection Module

The target detection module is designed to accurately pinpoint and extract the region of interest (ROI) from each slice image for precise measurements. Since the original images are large, approximately 2448×2048 pixels, and the ROI occupies only a small portion of the image, using smaller-sized images reduces the complexity and cost of training a model. Therefore, our approach involves training a You Only Look Once version 8 (YOLOv8) segmentation model to identify the ROI in the input image. This module selects a relevant section of the sliced image while eliminating unnecessary noise that occupies most of the area.

The ROI localization process has two main objectives. Firstly, we ensure all images are oriented in the same direction by using computer vision techniques to assist in alignment. Specifically, mask contours generated from YOLOv8 predictions are used to select small patches from both sides of the contour, and the average brightness of these patches determines whether the image and mask need to be flipped. subsequently, we proceed with precise ROI localization and cropping once the images and masks are aligned. Given the sparsity of the dataset,

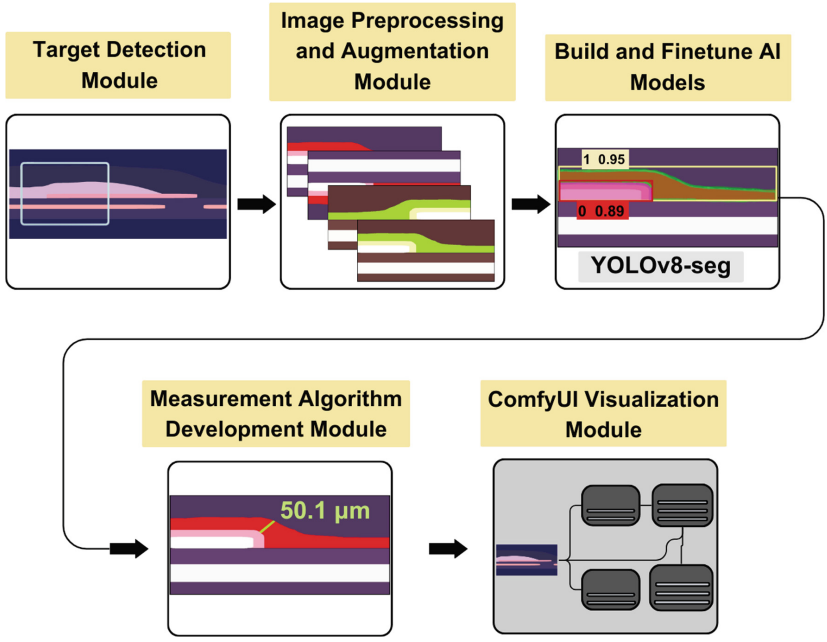

Fig. 3. The architecture of the proposed framework. In the section focused on building and fine-tuning AI models, the displayed image shows the results of YOLOv8 segmentation. Two bounding boxes are highlighted: the yellow box, labeled "1 0.95," indicates that the enclosed area has been segmented and classified as class 1 with a confidence score of 0.95. Similarly, the red box marks an area classified as class 0. The measurement result is 50.1 µm. Based on factory standards, we currently define 1 pixel as equivalent to 1 µm. (Color figure online)

minimizing image noise is crucial to prevent it from affecting the model's error rate, thereby helping the model capture relevant features and enhancing its accuracy and robustness. Our dataset comprises two types of slices: silver paste and metal layer. The results are shown in Fig. 4.

3.2 Image Preprocessing and Augmentation Module

During the image preprocessing stage, we start by normalizing the height and width of the ROI images. As the images produced by the target detection module may exhibit variations due to slight differences in aspect ratios or the presence of silver paste and metal layers, resizing them to a uniform size poses the risk of losing important features and reducing model accuracy. To address this, we resize the ROI images to match the input size parameters (image size) required by the model. Subsequently, we annotate the resized ROI images using Labelme. These annotations are then converted into TXT files for model training. We pay close attention to details, especially in the corners with the presence of silver paste, ensuring precise training of the AI model for accurate measurements.

Fig. 4. Silver paste slice image after target detection module.

Given the limited dataset of around 100 factory-provided microscopic slice images, data augmentation becomes crucial for improving the model's image interpretation. We expand the dataset using Albumentations, employing techniques including Contrast Limited Adaptive Histogram Equalization (CLAHE) [24], Optical Distortion, and GaussNoise. However, we refrain from using image flipping or random cropping during augmentation, as these could potentially compromise model performance and lead to inaccurate measurements.

3.3 Build and Finetune AI Models

In our study, we utilize the You Only Look Once version 8 (YOLOv8) model, a cutting-edge deep learning framework designed for a variety of computer vision tasks. Developed by Ultralytics in January 2023 [10], YOLOv8 encompasses five scaled versions: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large). This versatile framework supports a wide array of vision tasks, including object detection, segmentation, pose estimation, tracking, and classification. [23] For our study, we utilize the pre-trained YOLOv8n segmentation model, which was further fine-tuned to align with our specific dataset and objectives.

Following the initial training, we refine the model by adjusting key parameters such as image size and learning rate, as these greatly influence the accuracy of the training process. During the inference phase, the model demonstrates the capability to detect small areas within images. However, without careful management, these small areas may introduce errors when converting the detections into masks for subsequent measurements. Besides, small regions have the potential to yield misleading results. To address this issue, we implement an area filtering technique by establishing a threshold, which discards regions below a certain size and retains only the primary cutting areas for more accurate measurements.

Furthermore, we have documented key training parameters used in our experiments as a reference. While optimal parameter values may vary depending on

the specific type of slice measurement model, we have identified a set of parameters that significantly impacted the overall model performance. In the following sections, we provide a detailed introduction to these critical parameters.

Image Size. To optimize our model's performance, we experiment with different image sizes, as the ideal dimensions depend heavily on the training data. Smaller image sizes tend to facilitate more complete segmentation and often yield the highest accuracy. However, they may compromise pixel density and introduce deviations from the original image, resulting in larger pixel errors. On the other hand, larger image sizes generally produce smaller errors, leading to more precise calculations, but may hinder the segmentation of target areas, potentially causing incomplete or failed segmentation. Thus, identifying an appropriate image size is crucial to balancing segmentation completeness and calculation accuracy, ultimately enhancing training performance.

Image Augmentation. In the image preprocessing and augmentation module, we implement various augmentation techniques to enhance model robustness and reduce overfitting on unseen datasets. However, we find it essential to avoid simultaneously applying flipping, scaling, and rotation, as these transformations distort the original image features, resulting in a loss of critical characteristics. In real PCB manufacturing scenarios, such extreme variations in image orientation or scale do not occur, so including heavily transformed images could potentially confuse the model.

As a result, we refrain from using these particular augmentations in data preprocessing and partially disable YOLOv8's built-in augmentations during training. We experiment with fully disabling and partially reducing augmentations. While fully disabling augmentations sometimes lead to gradient instability, training with no augmentations at all resulted in reduced generalization. Thus, we opt to lower the intensity of augmentations for training, balancing the model's stability and its ability to generalize.

3.4 Measurement Algorithm Development Module

In this measurement algorithm development module, we utilize the inference results from the YOLOv8 segmentation model, extracting masks and bounding boxes data to develop the algorithms. We divide the algorithm development into two parts: one focusing on the silver paste corner thickness and the other on the metal layer. Initially, the metal layer requires measurements across eight sections; however, some algorithms can be applied repeatedly to different regions. Therefore, we will concentrate on key segments to demonstrate the development and application of these algorithms.

Silver Paste Corner Thickness. The goal of the algorithm is to find the shortest distance from the top-right corner of the silver paste section. Two masks

are detected and segmented by the model in the previous stage. The process begins by locating the top-right corner of the first mask, which serves as a pivotal reference point. This corner is identified by analyzing the mask to find the point with the smallest y-coordinate, followed by selecting the point with the maximum x-coordinate among those with the same y-value. However, the edge where the silver paste contacts the copper is not always smooth-irregularities and bumps can lead to inaccuracies when identifying the top-right corner, potentially affecting the measurement results. To mitigate this issue, a height threshold is applied to the first mask (representing copper), converting any white pixels exceeding this threshold to black, thereby excluding them from the measurement range.

Once the top-right corner is accurately identified, a vertical line is extended upward to intersect with the contour of the second mask. The highest intersection point, where the vertical line meets the contour, is then identified. Using this intersection point, we identify 100 nearest adjacent vertices in the contour. Finally, we calculate the Euclidean distance between the top-right corner and these 100 points, determining the shortest line segment and its corresponding coordinates.

Metal Layer. Calculating the thickness of metal layers is a fundamental and widely used method in microsection measurement. The objective is to measure the thickness at specified locations of metal layers, such as the thickness at the center or the average thickness in the central area. This algorithm can be applied to calculate the thickness of any layer, not just metal layers, by simply replacing the mask.

The specific implementation involves using a 2D matrix mask to perform the calculation. The size of the mask corresponds to the image size after YOLOv8 scaling. In the mask, elements with a value of 0 indicate that the pixel at that coordinate does not belong to the target area, whereas non-zero values indicate that the pixel is part of the target area. The next step is to extract the columns from the mask that correspond to the x-coordinate of the target (e.g., the center point or the columns within 15% of the center for calculating the average thickness in the central 30%). The thickness parameter is then determined by counting the number of non-zero elements in these columns, adjusted according to the scaling factor between the original image size and the scaled image size processed by YOLOv8. Finally, by applying the scale factor for the actual length-to-pixel ratio, the target thickness in real-world units can be obtained.

It is important that this algorithm requires scaling based on the ratio between the original image and YOLOv8's processed image. As discussed in Sect. 3.3, the image size can significantly impact the accuracy of the final calculation. An overly small image will necessitate greater scaling when converting the result back to the original image's pixel count, thereby amplifying any minor calculation errors and leading to inaccurate results.

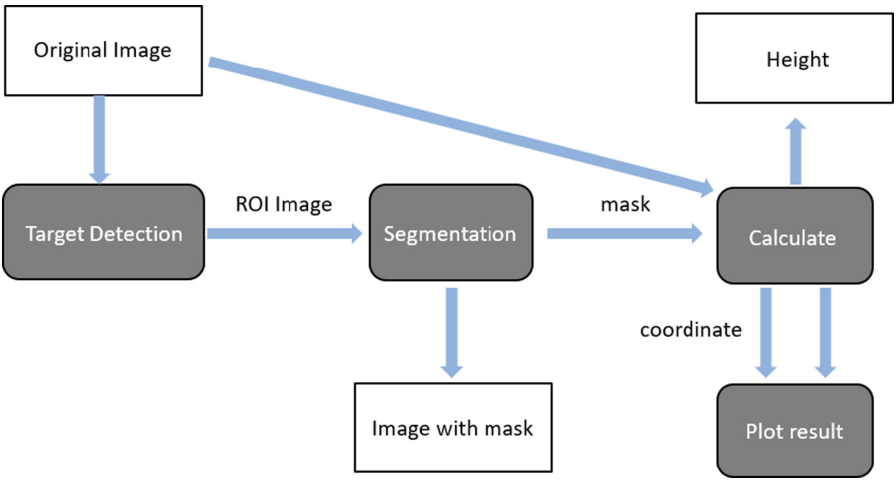

Fig. 5. Entire workflow within ComfyUI.

3.5 ComfyUI Visualization Module

In the ComfyUI Visualization Module, upon completing the microsection measurements and in response to factory requirements, we needed to establish a user interface that allows users to verify accuracy by annotating measurement line segments and coordinate points directly onto the original image. We selected ComfyUI for this task due to its versatility and user-friendly nature, making it well-suited for various AI models. ComfyUI also enhances visualization and offers flexibility through extensive configurability. Our approach involved integrating the Yolov8 model and the developed algorithms within ComfyUI and developing new nodes as needed.

During the development of the microsection measurement algorithm, we identified certain methods as similar or redundant, such as those related to the metal layer. To improve efficiency and code reusability, our study integrates ComfyUI to create a comprehensive user interface workflow for the entire measurement process. Setting up this workflow involves developing new nodes, adjusting node parameters such as variables, and establishing connections between nodes by linking their respective inputs and outputs.

ComfyUI. ComfyUI is originally designed for Stable Diffusion, featuring the ability to construct complex workflows using nodes. It supports customizable nodes and includes image preview functionality. After evaluation, we adopt it as a workflow design tool for microsection measurements. The workflow is illustrated in Fig. 5. The motivation for designing this workflow stems from the fact that the processes involved in microsection measurements are generally similar and repetitive. The only differences typically involve changes in models and slight adjustments to parameters. Utilizing a well-designed workflow can reduce

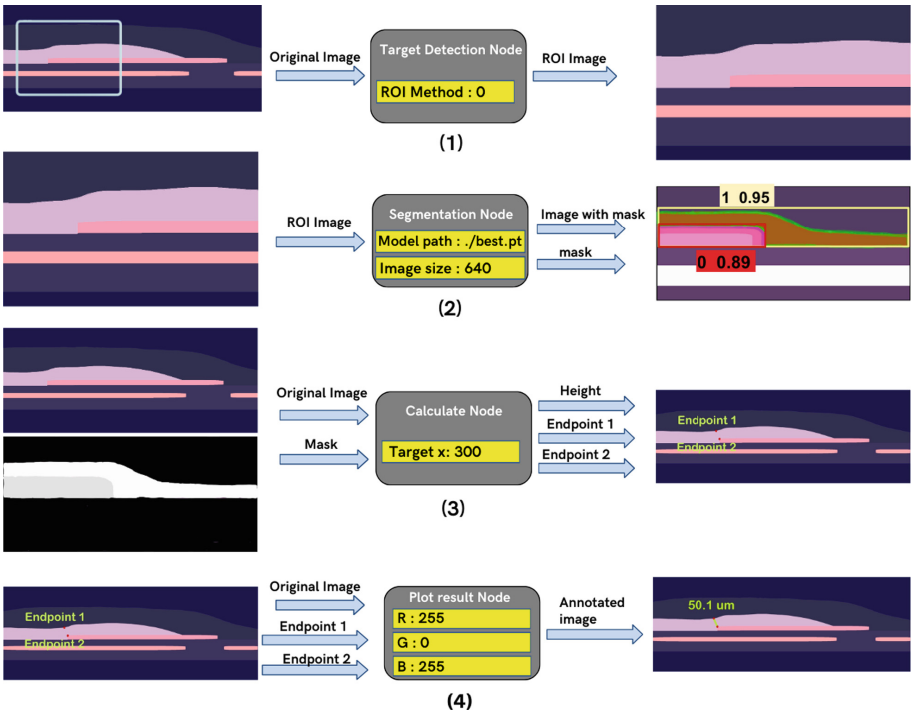

Fig. 6. Nodes of ComfyUI workflow. (1)target detection node (2) segmentation node (3) calculation node (4) plot result node. In the calculation node (3), it is important to note that the mask itself is composed of black-and-white values. However, to visually differentiate between classes, we use white as the mask for class 1 and grey for class 0. (Color figure online)

development costs. Additionally, designing the workflow with ComfyUI offers several advantages:

High Visualization. The node-based presentation makes the interaction between modules more intuitive and comprehensible. The image preview feature allows for a clear and immediate assessment of model accuracy and whether the locations measured are correct.

High Modularity. The nodes' versatility allows them to be used across different tasks. The nodes developed so far can be reused or slightly modified for microsection measurements, saving time and resources.

High Readability. The node-based approach, which abstracts the core functions into a black-box format, requires only an understanding of the input/output and the function's purpose, thus speeding up comprehension for others. A schematic of the nodes is shown in Fig. 6.

Ease of Use. The simple interface of ComfyUI makes it user-friendly, allowing it to be provided directly to clients for use.

Generate New Nodes. The default nodes in ComfyUI are primarily designed for text-to-image AI drawing, making them less suitable for use with YOLOv8. To address this, we developed several custom nodes tailored for slice measurement tasks. Most of these nodes are versatile and can be applied to various measurement tasks, eliminating the need for redevelopment when new measurement tasks arise in the future, thus saving significant time and resources. The key nodes are similar to the previously mentioned modules. Below is an introduction to each node, along with corresponding illustrations:

Target Detection Nodes. This node is linked to the Target Detection Module and processes the original image to produce a segmented output. To meet various segmentation needs, we've included common methods-brightness-based, color-based, and position-based-in a library, allowing users to select the appropriate option directly within the node.

Segmentation Node. In the segmentation node, the Yolov8 model weight is loaded to segment the target masks. The input for this node is the ROI image from the previous node, and the output is the inferred mask. Users have the flexibility to adjust various parameters within this node, such as the model path, image size, and the specific class ID of masks to be output.

Measurement Node. This calculation node corresponds to the Measurement Algorithm Development Module. It computes the designated distance and the coordinates of two endpoints based on the mask generated by the preceding node. The input comprises the mask and the x-coordinates to be measured, while the output includes the calculated distance and the coordinates of the two endpoints. This node can accommodate different algorithms depending on the specific task, providing users with the flexibility to switch and optimize workflows for different measurements.

Plot Result Node. This node visualizes the coordinates of the two endpoints calculated by the Calculate Node onto the image segmented by the Target Detection Node. We use Imagedraw to implement this node. It provides an intuitive visualization, enabling developers and users to quickly understand the exact measurement locations and lengths, which helps in assessing accuracy.

4 Experiments

In this section, we conduct experiments using two types of FPC board slices. The distances and coordinates that need to be calculated are illustrated in Fig. 1 and Fig. 2. We perform a comparative analysis between using the images processed by our target detection module as model input and using the original images.

Given that our measurement process is fundamentally dependent on masks, the precision of mask segmentation is identified as a pivotal factor, directly influencing the reliability of our results. Hence, we emphasize the evaluation metrics used to assess the performance of the trained model.

The experiments are conducted on two datasets: silver paste and metal layer, each initially containing 103 images. After augmenting both datasets fivefold, the total number of images in each dataset increases to 615. For each dataset, we allocate 80% for training, 10% for validation, and 10% for testing. The model is implemented using PyTorch as the backend framework, and YOLOv8 is trained on a workstation equipped with an NVIDIA GeForce RTX 4090 (24.56 GiB memory) and operated within a Jupyter Notebook environment.

4.1 Model Performance Evaluation

To evaluate the instance segmentation capabilities of YOLOv8 models, we employ four key metrics: Precision, Recall, and mean Average Precision (AP) from 0.5 to 0.95 intersection over union (mAP50-95 IOU). Precision, as defined in Eq. (1), represents the ratio of correctly identified positive instances to the total number of positive predictions. Recall, as described in Eq. (2), measures the proportion of correctly identified positive instances relative to the total actual instances of the target objects. Additionally, the mean Average Precision (mAP), calculated as the average AP across k categories Eq. (4), serves as a more rigorous and comprehensive metric for evaluating model performance, considering multiple overlap thresholds between predicted and actual object boundaries. Specifically, the mean average precision calculated at varying IoU thresholds from 0.50 to 0.95, with a stride of 0.05, provides a nuanced assessment of the model's precision under different degrees of overlap.

The evaluation results are summarized in Table 1, showcasing the strong performance of the four YOLOv8 models, with recall reaching 100% in all metal layer scenarios. Precision remains high, ranging from 0.99 to 0.998, and mAP50 values consistently exceed 0.993, confirming accuracy at the 50% IoU threshold. Even under the stricter mAP50-95 metric, results are robust, with the highest reaching 0.995. These results underscore the effectiveness of our models, as further demonstrated by the YOLOv8 segmentation in Table 1.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$IOU = \frac{AreaOverlap}{AreaUnion} = \frac{TP}{FP + TP + FN} \tag{3}$$

$$mAP = \frac{1}{k} \sum_{i=0}^{k} (AP)i \tag{4}$$

Table 1. The performance metrics of YOLOv8 models for two main scenarios.

Models	Precision	Recall	mAP50	mAP50-95
Silver paste	0.998	0.992	0.993	0.979
Metal layer (1)	0.996	1.0	0.995	0.995
Metal layer (2)	0.992	1.0	0.995	0.981
Metal layer (3)	0.99	1.0	0.995	0.90

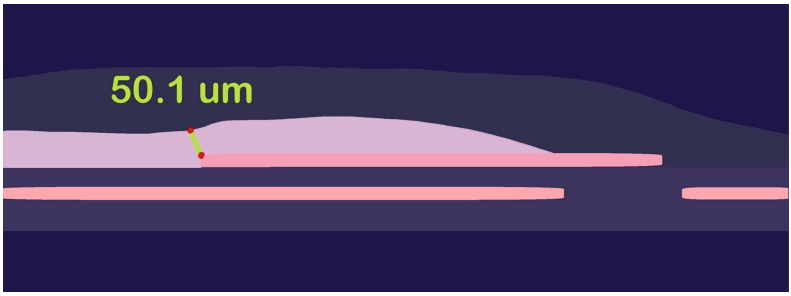

Fig. 7. Measurement result of silver paste corner thickness.

4.2 Target Measurement Results

After performing measurements using masks and annotating the resulting images, we simulate the real-world application scenario by using Microsoft Paint to assess measurement errors. The measurement result of silver paste Corner thickness is shown in Fig. 7. By magnifying the images to examine individual pixel blocks, we adhere to the stringent factory standard that mandates a measurement error margin of no more than 2 pixels. By employing masks for these measurements, we test a dataset of 200 images from the factory. Remarkably, we achieve a flawless measurement error of 0 pixels. This result demonstrates that using masks for measurement is an excellent choice, providing a high degree of accuracy. This result underscores that using masks for measurement is an exceptionally accurate method, which makes it highly reliable for tasks that demand precise accuracy.

5 Conclusion

In this paper, we presented an AI-based Microsection Measurement framework integrated with a ComfyUI workflow. To overcome challenges associated with limited data and low accuracy, we implemented a target detection module to minimize image noise and concentrate on the target measurement area. Furthermore, to achieve precision within 2 pixels, we developed a measurement method utilizing masks. This approach allowed us to achieve an accuracy of 0 pixels for

both silver paste and metal layers, significantly exceeding the 2-pixel requirement. Additionally, we have integrated all these functions into ComfyUI nodes and packaged the system in an executable form, enabling users to predict the width of elements without requiring a Python environment. The experiment results showed that we achieved a perfect measurement error of 0 pixels for 200 real-world images of factories. All in all, we not only provided a straightforward and user-friendly microsection measurement framework but also intensively increased the efficiency of microsection measurement.

Acknowledgements. This work is sponsored by the National Science and Technology Council (NSTC) under the project NSTC 113-2222-E-008-002 and NSTC 112-2622-8-A49-021.

References

1. Alsuwaylimi, A.A.: Enhanced YOLOv8-Seg instance segmentation for real-time submerged debris detection. IEEE Access **12**, 117833–117849 (2024)
2. Bai, R., Wang, M., Zhang, Z., Lu, J., Shen, F.: Automated construction site monitoring based on improved YOLOv8-Seg instance segmentation algorithm. IEEE Access **11**, 139082–139096 (2023)
3. Botero, U.J., et al.: Automated via detection for PCB reverse engineering. In: International Symposium for Testing and Failure Analysis, vol. 83348, pp. 157–171. ASM International (2020)
4. Charles, R., et al.: Master thesis: generative AI methods to create comic strips (2024)
5. Chen, I.C., Hwang, R.C., Huang, H.C.: PCB defect detection based on deep learning algorithm. Processes **11**(3), 775 (2023)
6. Chen, X., Wu, Y., He, X., Ming, W.: A comprehensive review of deep learning-based PCB defect detection. IEEE Access **11**, 139017–139038 (2023)
7. Girshick, R.: Fast R-CNN, pp. 1440–1448. IEEE, New York (2015)
8. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
9. Gung, J.J., Lin, C.Y., Lin, P.F., Chung, W.K.: An incremental meta defect detection system for printed circuit boards. In: 2022 IEEE International Conference on Consumer Electronics-Taiwan, pp. 307–308. IEEE (2022)
10. Jocher, G., Chaurasia, A., Qiu, J.Y.: By ultralytics (2023). https://github.com/ultralytics/ultralytics (2023)
11. Leong, W., Abdullah, M.Z., Khor, C.: Application of flexible printed circuit board (FPCB) in personal computer motherboards: focusing on mechanical performance. Microelectron. Reliab. **52**(4), 744–756 (2012)
12. Li, Y.Y., Li, Z.J., Zheng, H.X., Wang, C.R.: Comparison and implementation of image edge detection algorithm. Comput. Eng. Des. **31**(9), 1971–1975 (2010)
13. Lin, C.Y., et al.: A deep learning-based microsection measurement framework for print circuit boards. In: 2023 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), pp. 291–294. IEEE (2023)

14. Ling, Q., Isa, N.A.M.: Printed circuit board defect detection methods based on image processing, machine learning and deep learning: a survey. IEEE Access **11**, 15921–15944 (2023)
15. Luo, H., Deng, Z., Huang, H., Liu, X., Chen, R., Liu, Z.: VersusDebias: Universal zero-shot debiasing for text-to-image models via SLM-based prompt engineering and generative adversary. arXiv preprint arXiv:2407.19524 (2024)
16. Luo, W., Luo, J., Yang, Z.: FPC surface defect detection based on improved faster R-CNN with decoupled RPN. In: 2020 Chinese Automation Congress (CAC), pp. 7035–7039. IEEE (2020)
17. Malge, P., Nadaf, R.: PCB defect detection, classification and localization using mathematical morphology and image processing tools. Int. J. Comput. Appl. **87**(9), 40–45 (2014)
18. Park, J.H., Kim, Y.S., Seo, H., Cho, Y.J.: Analysis of training deep learning models for PCB defect detection. Sensors **23**(5), 2766 (2023)
19. Pérez-Colado, I.J., Freire-Morán, M., Calvo-Morata, A., Pérez-Colado, V.M., Fernández-Manjón, B.: AI asyet another tool in undergraduate student projects: Preliminary results. In: 2024 IEEE Global Engineering Education Conference (EDUCON), pp. 1–7. IEEE (2024)
20. Putera, S.I., Ibrahim, Z.: Printed circuit board defect detection using mathematical morphology and MATLAB image processing tools. In: 2010 2nd International Conference on Education Technology and Computer, vol. 5, pp. V5–359. IEEE (2010)
21. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
22. Sapkota, R., Ahmed, D., Karkee, M.: Comparing YOLOv8 and mask R-CNN for instance segmentation in complex orchard environments. Artif. Intell. Agric. **13**, 84–99 (2024)
23. Terven, J., Córdova-Esparza, D.M., Romero-González, J.A.: A comprehensive review of YOLO architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS. Mach. Learn. Knowl. Extr. **5**(4), 1680–1716 (2023)
24. Zuiderveld, K.: Contrast limited adaptive histogram equalization. In: Graphics gems IV, pp. 474–485 (1994)