A Dual-Strategy Framework for High-Resolution Typographic Defect Segmentation

1st Wenzheng Huang
School of Artificial Intelligence and Advanced Computing
Xi'an Jiaotong-Liverpool University
Suzhou, China
Wenzheng.Huang23@student.xjtlu.edu.cn

1st Zhanchong Liu
School of Artificial Intelligence and Advanced Computing
Xi'an Jiaotong-Liverpool University
Suzhou, China
Zhanchong.Liu23@student.xjtlu.edu.cn

2nd Yasunori Sakai*
School of Intelligent Manufacturing Ecosystem
Xi'an Jiaotong-Liverpool University
Suzhou, China
yasunori.sakai@xjtlu.edu.cn

3rd Jiachen Yuan
School of Artificial Intelligence and Advanced Computing
Xi'an Jiaotong-Liverpool University
Suzhou, China
Jiachen.Yuan23@student.xjtlu.edu.cn

Abstract. High-resolution typographic quality control is challenged by the detection of micro-defects, which are often minuscule and sparsely distributed, leading to a severe class imbalance problem. This paper proposes a novel end-to-end framework to address this issue. We introduce a dual-strategy for training, which combines data-level weighted patch sampling with a loss-level composite function comprising Weighted Cross-Entropy, Dice, and Focal losses. The segmentation network leverages a convolutional neural network (CNN) backbone with a semantic segmentation architecture to achieve robust feature extraction and precise localization. For practical deployment, we design a seamless inference pipeline using an overlapping sliding window to process ultra-high-resolution images without artifacts. Furthermore, an application-specific post-processing step generates intuitive visualizations for industrial quality inspection. Our experiments demonstrate that this comprehensive approach significantly improves the detection accuracy of typographic defects, providing a robust and practical solution for automated quality assurance.

Keywords: Semantic Segmentation, Typographic Defect, Class Imbalance, U-Net++, ResNet, High-Resolution Imaging, CVAT

I. Introduction

In the field of printed circuit board (PCB) inspection, printed text and defect recognition are not only crucial tasks to ensure product reliability and quality, but also represent a typical application scenario of high-resolution industrial image inspection. Printed text is often used for component processing and positioning. Early industrial methods mainly relied on optical character recognition (OCR) and traditional image processing algorithms, where researchers typically adopted binarization, projection, and template matching techniques to identify characters printed on PCB components in pursuit of efficient and high-quality solutions. However, although these methods performed well under controlled environments, they lacked robustness in practical applications, being extremely sensitive to illumination variations, background noise, and font style changes. More critically, binarization severely compresses image features, making it insufficient for fine-grained defect detection in high-resolution settings [1] [2] [3].

With the advancement of deep learning, the introduction of convolutional neural network (CNN) models has significantly enhanced the ability to recognize characters and markings in high-resolution images. For example, Jia and Liu (2022) proposed the ICChaNet model, which achieved an accuracy of over 98% in integrated circuit character recognition tasks, demonstrating the potential of improved CNNs in PCB character detection [5]. Meanwhile, CNN- and YOLO-based approaches have also been applied to tasks such as missing components, solder joint defects, and misaligned wiring. This indicates a paradigm shift in inspection technologies—from relying on pre-designed handcrafted features to adopting end-to-end trainable models—greatly improving flexibility and applicability [4] [6].

Nevertheless, current research still focuses primarily on device-level and character-level defect detection, while pixel-level fine-grained inspection of silkscreen printing remains insufficient. In industrial production, issues such as stroke interruptions, mouse-bite defects, and stroke deformations still occur frequently. Yet existing methods often provide only coarse assessments at the component level (OCR), or use bounding boxes to highlight defective character regions (CNN, YOLO), without enabling precise localization of defects in high-resolution images. This gap serves as the motivation for our study.

To address this limitation, this paper proposes a pixel-level defect detection system for PCB silkscreen printing. The core idea is to leverage the powerful feature representation of CNNs together with the precise localization capability of semantic segmentation networks, in order to detect minute defects in high-resolution images. Specifically: (i) we design a hybrid segmentation architecture that employs ResNet-101(CNNs) as the encoder for feature extraction and U-Net++(semantic segmentation networks) as the decoder for pixel-level localization, enabling high-fidelity detection of subtle printing defects; (ii) to address severe data imbalance caused by large background noise, we introduce a dual optimization strategy combining defect-aware class weighting and a composite loss function, thereby enhancing the model's ability to learn from rare defects; (iii) for ultra-high-resolution industrial images, we devise a seamless inference pipeline based on sliding-window cropping, which effectively improves the perception of small targets; (iv) finally, we construct an end-to-end defect-oriented framework that not only outputs pixel-level masks, but also generates intuitive reports through region masking and boundary visualization, thus better serving industrial quality inspection workflows.

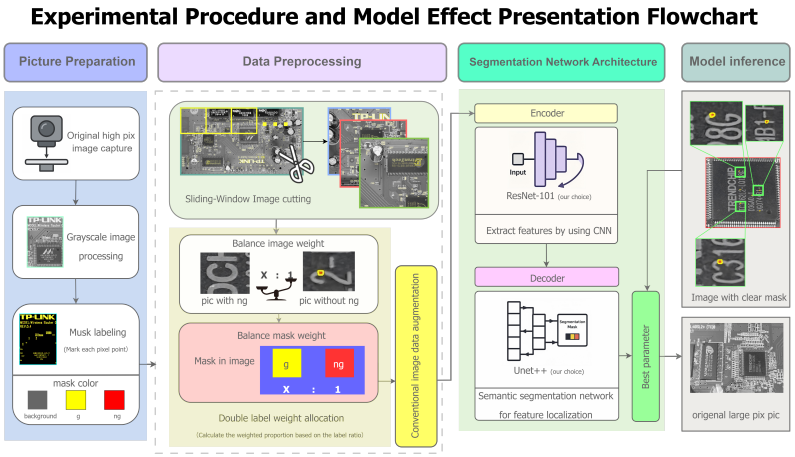


Fig. 1. The overall architecture of our proposed framework. The pipeline consists of two main stages: (Left) The training and validation pipeline, which incorporates our dual-strategy for handling class imbalance with the ResNet-101 and U-Net++ based network. (Right) The inference and deployment pipeline, featuring overlapping sliding window inference and application-specific post-processing for generating final defect reports.

II. Methodology

To address the challenges of detecting micro-defects in high-resolution typographic images, we propose a comprehensive framework as illustrated in Fig. 1. Our methodology encompasses a meticulous data preparation workflow, a powerful segmentation network architecture, a dual-strategy training regimen, and a practical inference pipeline.

A. Image Preparation and Data Preprocessing

The creation of a high-quality, standardized dataset is the foundational step for successful supervised learning. This study uses a dataset consisting of 13 high-resolution printed images, in which the defect areas account for approximately 0.1% of all pixels, exhibiting a highly imbalanced distribution. Our pipeline is mainly divided into two stages: Image Standardization and pixel-level detailed annotation.

1) Image Standardization: The raw source images for this study vary in format and may contain color information irrelevant to the structural nature of typographic defects. To create a uniform dataset, we first employ an automated Python script to perform batch processing. This script reads all images from an input directory, converts them to 8-bit single-channel grayscale using the OpenCV library, and saves them to a designated ‘images/‘ folder. This standardization ensures that the model focuses exclusively on morphological and textural features, rather than color, and provides a consistent input format for the subsequent stages.The choice of 8-bit single-channel grayscale images is made because it preserves sufficient texture details while significantly reducing computational complexity, making it well-suited for processing high-resolution images.

2) Pixel-Level Annotation and Mask Generation: To generate the pixel-wise ground truth required for semantic segmentation, we utilize the Computer Vision Annotation Tool (CVAT). Each standardized grayscale image from the ‘images/‘ folder is meticulously annotated by human researchers. The annotation process involves assigning each pixel to one of three predefined semantic classes: ‘defect area‘ (visualized in red for the annotator), ‘normal stroke‘ (visualized in yellow), and ‘background‘ (visualized in gray).

Upon completion of the annotation, the labels are exported using CVAT’s ”Segmentation Mask 1.1” format. This format generates a single-channel, 8-bit PNG image for each source image, where the color-coded annotations are converted into standardized integer pixel intensity values corresponding to each class (e.g., 0 for background, 1 for normal stroke, 2 for defect). These generated masks are stored in a ‘masks/‘ folder, ensuring a direct, one-to-one filename correspondence with the grayscale images. This structured dataset, with perfect semantic alignment between source images and their masks, forms the direct input to our training pipeline.

B. Segmentation Network Architecture

Our network design, based on the proven encoder-decoder paradigm, is tailored to the standardized dataset to achieve precise defect detection through an efficient segmentation network architecture.We select a powerful pre-trained ResNet-101 as the encoder for robust feature extraction and a U-Net++ decoder for precise defect localization through enhanced feature fusion.

1) Encoder · ResNet-101 with Residual Learning: Since deeper backbones like ResNet-101 have been shown to capture richer features and achieve superior segmentation performance compared to shallower variants [7], we adopt ResNet-101 as the encoder backbone of UNet++ in this study.To extract a rich hierarchy of features, we employ a ResNet-101 backbone, pre-trained on ImageNet. The core innovation of ResNet is the residual block, which mitigates the vanishing gradient problem. Instead of learning a direct mapping $\mathcal{H}(x)$, a residual block learns a residual function $\mathcal{F}(x, \{W_i\}) = \mathcal{H}(x) - x$. The output of the block is thus formulated as:

$$y = \sigma(\mathcal{F}(x, \{W_i\}) + x) \tag{1}$$

where x and y are the input and output vectors, and $\mathcal{F}(x, \{W_i\})$ is the residual mapping. The multi-scale feature maps from the ResNet-101 encoder serve as inputs to the U-Net++ decoder.

2) Decoder · U-Net++ with Nested and Dense Skip Connections: U-Net++ has been shown to significantly improve the detection of micro-defects in high-resolution silk screen printing images, especially when combined with patch-based input, achieving about 10% improvement in Dice score over conventional methods [8].So the U-Net++ architecture is utilized as the decoder to effectively fuse features from different semantic levels. Let $X^{i,j}$ denote the output of node (i, j),

where i is the down-sampling level and j is the convolutional layer on the skip pathway. The computation at each node is:

$$X^{i,j} = \begin{cases} \mathcal{H}(\mathcal{D}(X^{i-1,j})) & , \text{if } j = 0 \\ \mathcal{H}\left(\left[[X^{i,k}]_{k=0}^{j-1}, \mathcal{U}(X^{i+1,j-1})\right]\right) & , \text{if } j > 0 \end{cases} \quad (2)$$

where $\mathcal{H}(\cdot)$ is a convolution block, $\mathcal{D}(\cdot)$ and $\mathcal{U}(\cdot)$ are down/up-sampling operations, and $[\cdot]$ is concatenation. This dense connectivity enables a more flexible feature fusion.This formulation fuses multi-scale features through dense skip connections, significantly improving the localization accuracy of tiny defects, particularly in complex printed texture backgrounds.

C. Core Training Strategy: A Dual-Pronged Approach to Imbalance

To effectively train our network on the highly imbalanced typographic dataset, we developed a dual-pronged strategy that addresses the problem at both the data and loss levels.

1) Data-Level Balancing · Weighted Patch Sampling: After pre-slicing the high-resolution images and their masks into N patches of size 512×512, we employ a Weighted Random Sampler. For each patch p_i in the dataset $D = \{p_1, p_2, ..., p_N\}$, we assign a weight w_i. If a patch p_i contains at least one pixel of the 'defect_area' class, its weight w_i is set to a high value W_{defect} (e.g., 10); otherwise, $w_i = 1$. The probability $P(p_i)$ of sampling patch p_i during training is then defined as:

$$P(p_i) = \frac{w_i}{\sum_{j=1}^{N} w_j} \quad (3)$$

This strategy ensures that the model is frequently exposed to the rare but critical defect samples in every training batch, preventing the learning process from being dominated by the background class.

2) Loss-Level Optimization: Composite Weighted Loss Function: Complementing the data-level approach, we designed a composite loss function, L_{total}, which provides a more robust and targeted optimization objective. It is a weighted linear combination of three distinct loss functions:

$$L_{total} = \lambda_1 L_{WCE} + \lambda_2 L_{Dice} + \lambda_3 L_{Focal} \quad (4)$$

The hyperparameters $\lambda_1, \lambda_2, \lambda_3$ balance the contribution of each component. Their values were determined through empirical validation on our dataset, guided by the specific role each loss function plays in the optimization process. In our study, we set them to 0.3, 0.5, and 0.2, respectively.

The rationale behind this weighting scheme is strategic. We assign the highest weight ($\lambda_2 = 0.5$) to the Dice Loss, as it directly optimizes for spatial overlap (IoU), which is our primary evaluation metric and is inherently robust to class imbalance. The Weighted Cross-Entropy is given the next highest weight ($\lambda_1 = 0.3$) to enforce pixel-level classification accuracy, serving as a strong complement to the region-based Dice Loss. Finally, the Focal Loss is assigned the lowest weight ($\lambda_3 = 0.2$) to act as a fine-tuning mechanism, gently guiding the model to focus on hard-to-classify examples, such as ambiguous boundaries, without disrupting the stable learning provided by the other two components. Each component is detailed below.

a) Weighted Cross-Entropy (WCE): To directly counteract the class imbalance, we utilize a weighted version of the standard cross-entropy loss. For a given pixel i, where $y_{i,k}$ is a binary indicator if class k is the correct classification and $p_{i,k}$ is the model's predicted probability for that class, the WCE loss is formulated as:

$$L_{WCE} = -\frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{C} \alpha_k y_{i,k} \log(p_{i,k}) \quad (5)$$

Here, M is the total number of pixels in a patch and C is the number of classes. The key component is the class weight α_k. We assign a very high weight to the 'defect_area' class ($\alpha_{defect} = 100$), thereby imposing a substantial penalty for misclassifying any defect pixel and forcing the model to prioritize their correct identification.

b) Dice Loss: The Dice Loss is a region-based metric that is inherently robust to class imbalance as it measures the overlap between the predicted and ground truth segmentation masks. It directly optimizes the Dice Coefficient, which is closely related to the IoU metric. For a multi-class scenario, the Dice Loss is defined as:

$$L_{Dice} = 1 - \frac{1}{C} \sum_{k=1}^{C} \frac{2 \sum_{i=1}^{M} p_{i,k} y_{i,k} + \epsilon}{\sum_{i=1}^{M} p_{i,k} + \sum_{i=1}^{M} y_{i,k} + \epsilon} \quad (6)$$

where ϵ is a small smoothing constant (e.g., 10^{-6}) to prevent division by zero. By minimizing this loss, the model is encouraged to produce spatially coherent and complete segmentation masks, which is particularly beneficial for capturing the full extent of defect regions.

c) Focal Loss: The Focal Loss was designed to address scenarios where there is a large imbalance between easy and hard examples. It dynamically down-weights the loss assigned to well-classified pixels, allowing the model to focus its learning efforts on the more challenging, hard-to-classify pixels (such as those at the ambiguous boundaries of defects). It achieves this by adding a modulating factor $(1 - p_{i,k})^\gamma$ to the standard cross-entropy loss:

$$L_{Focal} = -\frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{C} \alpha_k (1 - p_{i,k})^\gamma y_{i,k} \log(p_{i,k}) \quad (7)$$

where γ is the focusing parameter, typically set to 2. When a pixel is classified with high confidence (i.e., $p_{i,k}$ is close to 1), the modulating factor approaches 0, reducing that pixel's contribution to the total loss. This prevents the vast number of easy background samples from overwhelming the learning process.

D. Inference and Deployment Pipeline

1) High-Resolution Inference: Overlapping Sliding Window: For oversized images, we use an overlapping sliding window. The final logits vector for a pixel at (x, y), $L_{final}(x, y)$, is the average of predictions from all K windows covering it:

$$L_{final}(x, y) = \frac{\sum_{k=1}^{K} \mathbb{I}((x, y) \in W_k) \cdot L_k(x, y)}{\sum_{k=1}^{K} \mathbb{I}((x, y) \in W_k)} \quad (8)$$

where W_k is the area of the k-th window and $\mathbb{I}(\cdot)$ is the indicator function. This logit-level averaging eliminates stitching artifacts.

2) Application-Oriented Post-Processing: The final stage translates the model's output into actionable feedback. The prediction map is binarized to isolate the 'defect area' class. This mask is then used to generate a dual-enhanced visualization on the original image, featuring a semi-transparent red overlay and a high-contrast contour.

III. Experiments

In this section, we conduct a series of comprehensive experiments to validate the efficacy of our proposed framework. We first describe the experimental setup and evaluation metrics, followed by ablation studies and comparative analyses to systematically evaluate our design choices. Finally, we present qualitative results and a case study to demonstrate the practical performance and generalization capabilities of our system.

A. Experimental Setup

a) Dataset: The experiments were conducted on a proprietary dataset of 13 high-resolution typographic images, with an average size of approximately [3252x3965] pixels. The dataset was randomly partitioned into training and validation sets following an 80:20 ratio. All images and masks were prepared according to the workflow detailed in Section III-A.

b) Implementation Details: The proposed framework was implemented using PyTorch and the segmentation-models-pytorch library. All experiments were executed on a workstation equipped with an [NVIDIA RTX 3090 24GB] GPU. For training, high-resolution images were pre-sliced into patches. We utilized the Adam optimizer with an initial learning rate of 1×10^{-4} and employed the CosineAnnealingWarmRestarts scheduler. All models were trained for 200 epochs with a batch size of 6. The training process is visualized in Figure 2, which illustrates the model's stable convergence over the training duration.

B. Evaluation Metrics

To quantitatively evaluate model performance, especially under the severe class imbalance inherent in our task, we adopt the Intersection over Union (IoU), also known as the Jaccard index, as our primary metric. IoU measures the ratio of the intersection to the union of the predicted segmentation and the ground truth mask. We report two specific metrics: (1) Mean IoU (mIoU): The average IoU across all classes, reflecting the overall segmentation quality. (2) Per-class IoU: We place particular emphasis on the IoU for the defect_area class, as it directly quantifies the model's ability to solve the core problem of this study.

C. Performance Evaluation and Analysis

1) Architecture Comparison: To validate our choice of the U-Net++ architecture, we compared it against two other widely-used segmentation models: U-Net and DeepLabV3+. To ensure a fair comparison, all models employed the same ResNet-101 encoder and were trained using our proposed dual-strategy for imbalance handling (i.e., weighted sampling and the composite loss function).

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT SEGMENTATION ARCHITECTURES (IOU %)

Architecture Head	**Performance Metrics**			
	Mean	***Defect***	***Normal***	***Background***
U-Net	69.17%	34.29%	74.79%	98.43%
DeepLabV3+	67.70%	29.89%	74.81%	98.40%
U-Net++ [Ours]	**69.67%**	**34.41%**	**76.09%**	**98.52%**

The results of the architectural comparison are presented in Table I. The data indicates that both U-Net and U-Net++ significantly outperform DeepLabV3+ on the critical 'defect_area' IoU metric. This suggests that the U-Net family's encoder-decoder structure with skip connections is particularly well-suited for tasks requiring precise localization of fine-grained details, which DeepLabV3+'s atrous convolution-based spatial pyramid pooling may not capture as effectively.

While the performance of U-Net is competitive, our chosen U-Net++ architecture achieves the highest scores across all reported metrics, including the most crucial 'defect_area' IoU (34.41%) and the overall mIoU (69.67%). The slight but consistent improvement over the standard U-Net can be attributed to U-Net++'s redesigned nested and dense skip pathways. These pathways allow for a more flexible and effective fusion of features from different semantic levels of the encoder, which is vital for accurately delineating the complex and often ambiguous boundaries of typographic defects. This confirms that U-Net++ is the most suitable architecture for this specific task.

2) Ablation Study on the Loss Function: To demonstrate the necessity of our composite loss function, we conducted a critical ablation study. We fixed the network architecture to the superior 'ResNet-101 + U-Net++' and maintained the weighted sampling strategy. We then compared the performance of our proposed Composite Loss (CL) against a baseline using a standard, unweighted Cross-Entropy (CE) loss.

TABLE II
ABLATION STUDY RESULTS FOR THE LOSS FUNCTION (IOU %)

Loss Function Head	**Performance Metrics**			
	Mean	***Defect***	***Normal***	***Background***
Baseline	67.59%	29.47%	74.79%	98.51%
Composite Loss	**69.67%**	**34.41%**	**76.09%**	**98.52%**

The results presented in Table II reveal a crucial insight into the challenges of this task. The baseline model, while achieving a deceptively high mIoU of 67.59%, exhibits poor performance on the critical 'defect area' class, with an IoU of only 29.47%. This is masked by the near-perfect IoU on the dominant background class (98.51%), indicating that a standard loss function fails to address the core problem effectively.

In contrast, our proposed Composite Loss, while only modestly improving the overall mIoU, achieves a 'defect area ' IoU of 34.41%. This represents a 4.94 point absolute increase and a significant 16.8% relative improvement over the baseline for the most challenging and important class. This targeted improvement is a strong indicator of our method's efficacy.

Furthermore, we observed that the baseline model's performance on the defect class plateaued early in the training process. This suggests that the standard CE loss, overwhelmed by gradients from easy background samples, allows the model to converge to a trivial local minimum where defects are largely ignored. Our composite loss—combining Weighted Cross-Entropy, Dice Loss, and Focal Loss provides a more robust and persistent gradient signal for the minority class. This enables the model to overcome the performance plateau and continue refining its predictions for difficult defect samples, validating that our composite loss function is a decisive component for success in this task.

3) Inference Strategy Comparison: The inference strategy is critical for generating high-quality, globally consistent segmentation maps from large images. To evaluate this, we conducted a comprehensive comparison between a baseline non-overlapping (0% overlap) sliding window and our proposed 50% overlap strategy. The evaluation was both quantitative, focusing on detection reliability across the entire image, and qualitative, illustrating the typical output of each method.

The quantitative analysis revealed a significant improvement in detection reliability. The baseline 0% overlap strategy yielded 28 total detections, of which 6 were identified as false positives (e.g., minor board depressions or insignificant character variations). In stark contrast, the 50% overlap strategy produced a more refined set of 25 detections, with only a single false positive.

This substantial reduction in false alarms (from 6 to 1) demonstrates a key advantage of the overlapping approach. We attribute this to the increased contextual information and the corrective nature of our logit-level fusion process. Features that may appear ambiguous at the edge of a non-overlapped patch are viewed multiple times with greater surrounding

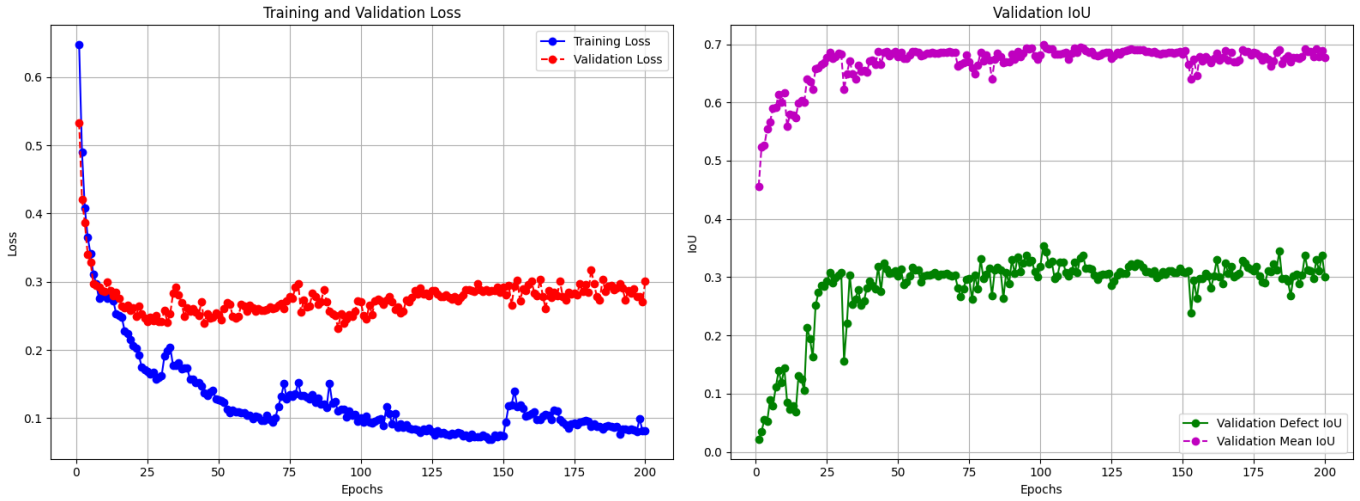


Fig. 2. Training and validation curves of the proposed model. (Left) Training loss (blue) and validation loss (red) across 200 epochs. The periodic jumps in the training loss correspond to the restarts of the Cosine Annealing scheduler. (Right) Validation Intersection over Union (IoU) for the 'defect area' class (green) and the mean IoU (purple). The steady increase and stabilization of the IoU curves indicate successful model learning and convergence on the validation set.

context in the 50% overlap scenario. The averaging of these multiple, more confident predictions effectively suppresses spurious detections that might arise from an incomplete view.

Fig. 3 provides a visual example of the final outputs that were subjected to this quantitative analysis.

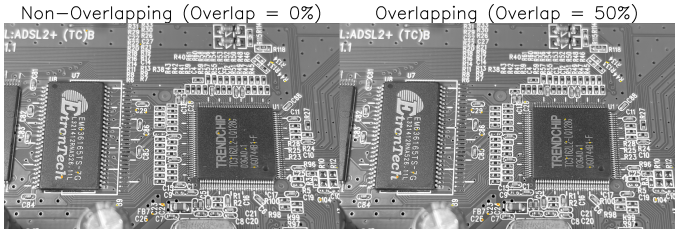


Fig. 3. Visual illustration of the final reports generated by the two inference strategies. Although the full set of 28 and 25 detections are distributed across the entire high-resolution image, this cropped example shows the typical visual quality of the output from the non-overlapping method (left) and our proposed overlapping method (right).

While the baseline approach is computationally efficient, its susceptibility to producing less reliable detections at patch boundaries makes it less suitable for high-stakes industrial inspection. The significant reduction in the false positive rate confirms that for tasks requiring high-fidelity localization, our proposed 50% overlapping sliding window is a crucial component for achieving robust and trustworthy results.

D. Qualitative Analysis and Case Study

To showcase the generalization capability and practical utility of our full framework, we tested it on a novel high-resolution image not seen during training or validation. Fig. 4 demonstrates that our system successfully identifies a very subtle ink spot defect. The pipeline clearly shows the progression from the model's generated binary mask (c) to the final, intuitive quality report with highlighted annotations (d). This result confirms that our method not only performs well on academic metrics but also holds significant potential for deployment in real-world industrial scenarios.

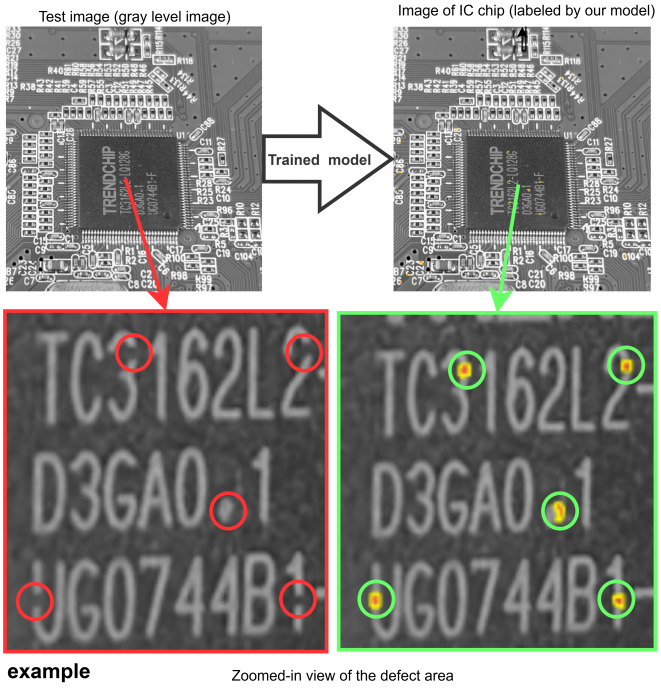


Fig. 4. Detection results on a novel test image. Top-left: original input; top-right: predicted binary mask. Bottom-left: zoomed-in defect region; bottom-right: final visualization with red overlay and yellow contour. The arrow indicates the transformation from input to mask.

IV. Discussion

The experimental results presented in Section IV systematically validate the effectiveness of our proposed framework for high-resolution typographic defect segmentation. Our findings offer several key insights into the challenges of this task and the efficacy of our methodological choices.

The architecture comparison in Table I clearly demonstrates that while modern encoder-decoder structures are generally capable, the choice of the decoder plays a critical role

in performance. The superior results of U-Net++, particularly on the ‘defect_area‘ IoU, underscore the importance of its dense and nested skip connections. We posit that this enhanced feature fusion mechanism is particularly adept at handling the fine-grained and often subtle morphological variations of typographic defects, enabling more precise boundary localization than standard U-Net or DeepLabV3+ architectures.

Perhaps the most significant finding is the dramatic performance gain revealed by the loss function ablation study (Table II). The baseline model, using a standard Cross-Entropy loss, was largely unable to overcome the severe class imbalance. The success of our composite loss function demonstrates the power of a multi-faceted optimization strategy: the Weighted Cross-Entropy directly penalizes errors on the rare class, the Dice Loss enforces regional consistency, and the Focal Loss focuses the model's attention on hard-to-classify boundary pixels. The synergy between these components proved to be decisive.

It is crucial to interpret the resulting ‘defect_area‘ IoU of 34.41% within the context of the practical industrial application. While this value may seem numerically moderate, its significance lies in the model's primary function: **defect localization**. An IoU of this magnitude indicates that the model is successfully identifying the presence and general location of true defects, which is the most critical first step in any quality control pipeline. Crucially, as suggested by the near-perfect background IoU (98.52%) and confirmed by qualitative analysis, the model exhibits a very low rate of false positives; it rarely misclassifies correct characters as defective. The primary reason for the IoU not being higher is not a failure to locate the defect, but rather an imprecision in delineating its exact pixel-perfect boundary. This boundary imprecision is a direct and expected consequence of our limited training dataset. Therefore, we argue that this IoU is a highly valuable and promising result, as it confirms the model's viability for the core industrial task of defect flagging and establishes a strong baseline.

Furthermore, the qualitative comparison of inference strategies (Fig. 3) provides a crucial practical insight. While often overlooked in academic studies, the generation of artifact-free predictions on large-scale images is paramount for deployment. Our results confirm that a non-overlapping sliding window is inadequate, as it fails to preserve spatial continuity across patch boundaries. The smooth, coherent predictions from our overlapping window strategy validate its necessity for producing reliable and visually coherent segmentation maps in a real-world setting.

Despite the strong performance, we acknowledge certain limitations which we frame as avenues for future work rather than fundamental problems with the approach. The current framework's performance is, as discussed, contingent on the limited quality and diversity of the annotated training data. We observed that the model sometimes misclassifies small, naturally occurring voids in the substrate as defects. We hypothesize that this is a limitation of the dataset's scale, which prevents the model from learning the subtle contextual cues to differentiate them from true defects.

Future work should therefore proceed in several directions. To enhance generalization and improve boundary precision, the training dataset must be expanded with more varied defect classes and hard-negative examples. For applications requiring faster inference, research into model compression techniques, such as knowledge distillation or quantization, could be beneficial. Finally, integrating the system's output with automated factory control systems to trigger real-time actions, such as part rejection or process alerts, would represent a significant step towards a fully automated quality assurance pipeline.

V. Conclusion

In this paper, we proposed and validated a comprehensive, end-to-end framework for the segmentation of micro-defects in high-resolution typographic images. Our methodology was specifically designed to overcome the dual challenges of severe class imbalance and the practicalities of industrial deployment. The framework's architecture synergistically combines a deep ResNet-101 encoder for robust feature extraction with an advanced U-Net++ decoder for precise localization.

Our primary contribution lies in the targeted optimization strategies that demonstrably enhance performance on this difficult task. Through a series of rigorous experiments, we quantitatively validated the superiority of each design choice. Our architectural comparison confirmed that the U-Net++ based model achieved the highest performance, yielding a final ‘defect area‘ IoU of 34.41%. More critically, our ablation study on the loss function revealed the profound impact of our optimization strategy. By implementing a composite loss function, we increased the ‘defect area‘ IoU from a baseline of 29.47% to 34.41%, which constitutes a significant 16.8% relative improvement on the most critical and challenging class. This confirms that our dual-strategy—combining data-level sampling with a multi-component, loss-level optimization—is decisive for success.

Furthermore, our qualitative analysis demonstrated that the proposed overlapping sliding window inference strategy is essential for practical application, successfully eliminating the stitching artifacts produced by baseline methods. Ultimately, this work provides a robust and validated solution that effectively bridges the gap between theoretical deep learning models and the demanding requirements of industrial quality control. The presented framework establishes a strong baseline and methodology, paving the way for more reliable and efficient automated inspection systems in the printing and manufacturing industries.

References

[1] Y. Iano, D. K. Bonello, and U. B. Neto, “Text recognition in PCBs: An object character recognition (OCR) algorithm,” in *International Journal of Development Research*, vol. 10, no. 07, pp. 38650–38656, Jul. 2020.

[2] W. Li, S. Neullens, M. Breier, M. Bosling, T. Pretz, and D. Merhof, “Text recognition for information retrieval in images of printed circuit boards,” in *Proceedings of IEEE*, RWTH Aachen University, Germany, 2019.

[3] C.-H. Lin, S.-H. Wang, and C.-J. Lin, “Using convolutional neural networks for character verification on integrated circuit components of printed circuit boards,” in *Applied Intelligence*, vol. 49, pp. 4022–4032, May 2019.

[4] S. Ghosh, M. A. M. Sathiaseelan, and N. Asadizanjani, “Deep learning-based approaches for text recognition in PCB optical inspection: A survey,” in *IEEE Physical Assurance and Inspection of Electronics (PAINE)*, 2021, pp. 1–8.

[5] X. Jia and Z. Liu, “Character identification for integrated circuit components on printed circuit boards using deep learning,” in *Journal of Electrical Engineering & Technology*, vol. 17, pp. 601–616, Sep. 2021.

[6] J. Tang, S. Liu, D. Zhao, L. Tang, W. Zou, and B. Zheng, “PCB-YOLO: An improved detection algorithm of PCB surface defects based on YOLOv5,” in *Sustainability*, vol. 15, no. 7, p. 5963, Mar. 2023.

[7] C. Kamann and C. Rother, “Benchmarking the robustness of semantic segmentation models,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 8828–8838.

[8] Y. Yu, Y. Jiang, and C. Zhang, “Improved U-Net++ with Patch Split for Micro-Defect Inspection in Silk Screen Printing,” in *Applied Sciences*, vol. 12, no. 9, p. 4679, 2022.