ML4Science Project Report: Detection of Tiny Critical Defects in Superconducting Circuits

Fedor Chikhachev[1], João Pinto[1], Pedro Pinto[1]

Hybrid Quantum Circuits Laboratory (HQC)
[1]*École Polytechnique Fédérale de Lausanne (EPFL), Switzerland*
CS-433 Machine Learning

***Abstract*—This project addresses the detection of critical defects in high-resolution microscopy images of superconducting circuits using deep learning-based object detection. Due to strict data privacy constraints, experiments were conducted locally by fine-tuning pre-trained models on a limited, manually labelled dataset. The task is particularly challenging due to minute defect sizes, significant class imbalance, and initial annotation inconsistencies. To overcome these obstacles, we iteratively refined the problem scope, improved labelling quality, and integrated Slicing Aided Hyper Inference (SAHI) to enhance small-object detection. Our final solution, based on a modified YOLO architecture, significantly improves performance from a baseline mean F1-score of 0.110 to a final score of 0.497, representing a nearly fivefold improvement with practically zero confusion between defect classes. These results underscore the importance of data quality and specialised inference strategies over raw architectural complexity in low-data industrial settings.**

I. Introduction

Superconducting circuits are at the heart of modern quantum technologies, enabling the realisation of high-coherence qubits and ultra-sensitive detectors. However, the scalability and performance of these devices are often limited by microscopic defects introduced during fabrication. These defects, ranging from lithographic imperfections to material inhomogeneities, can significantly impact critical parameters such as coherence time, critical current, and resonator quality factors. Currently, inspection workflows at the Center of MicroNanoTechnology (CMi) at EPFL are based on manual review of microscopy images. This process is time-consuming, subjective, and prone to missing subtle defect patterns. While leveraging Machine Learning (ML) offers a promising path to automate detection and improve fabrication protocols, applying standard computer vision models to this domain presents unique challenges.

The primary difficulty lies in the extreme scale mismatch between the input images and the target objects. The dataset consists of high-resolution microscopy images (2880×2160 pixels), whereas the defects of interest are often as small as 4-10 pixels in diameter. Standard object detection pipelines, such as the baseline YOLO architecture, typically rely on downsampling operations that suppress features of such small objects, making them virtually undetectable when processing the full image. Furthermore, strict data privacy constraints necessitated that all experimentation be conducted locally on personal hardware, precluding the use of large-scale cloud training resources or massive external datasets.

In this work, we propose a specialised ML pipeline for the identification of critical defects in superconducting microchips. To address the scale mismatch, we combine a fine-tuned YOLO-based detector [1] with Slicing Aided Hyper Inference (SAHI [2]). This approach allows for the preservation of fine-grained spatial details without the computational cost of processing ultra-high-resolution tensors directly. The project involved iterative refinement of the problem scope, extensive data cleaning, and the development of a data-centric workflow to maximise the utility of a limited, manually labelled dataset.

II. Related Work

Automated defect detection in high-resolution industrial imagery has been widely studied using convolutional neural networks, with recent approaches increasingly adopting object detection frameworks to enable both localisation and classification. Single-stage detectors from the YOLO family have demonstrated a favourable trade-off between detection accuracy and inference speed, making them suitable for real-time and resource-constrained industrial settings [1]. However, detecting small objects remains challenging for standard detection pipelines, as aggressive downsampling in deep networks often suppresses fine-grained features, particularly when objects occupy only a few pixels in large images [3].

To address this limitation, inference-level adaptations such as tiling or slicing have been proposed, allowing detectors to operate on smaller image regions while preserving spatial detail. Slicing Aided Hyper Inference (SAHI) [2] formalises this approach by partitioning large images into overlapping tiles and merging predictions in the original coordinate space, significantly improving recall for small objects without requiring architectural modifications. Motivated by these findings, our work combines a YOLO-based detector with

SAHI to address the challenges of small defect detection under strict data and computational constraints.

III. Dataset and Preprocessing

A. Dataset Characteristics

The dataset consists of high-resolution microchip images acquired in a laboratory environment using a Keyence VHX-X1 digital microscope. As discussed in Section I, the extreme object–image scale mismatch complicates automated detection and motivates preserving spatial detail during training and inference.

No annotated dataset was available at the start of the project. All labels were therefore created manually by the authors using Label Studio [4]. Initial annotations followed a multi-class scheme (*Dirt*, *Dirt-Wire*, *Burn*, *Open*); however, early experiments revealed that ambiguous class definitions and limited samples per category hindered effective learning.

B. Dataset Refinement and Preparation

Exploratory experiments highlighted inherent challenges, including the minute scale of defects, high visual variability across chip layouts, severe class imbalance, and inconsistent initial annotations. To address these issues, the project scope was iteratively refined. While an intermediate binary formulation distinguishing *Critical* from *Non-Critical* defects was evaluated, expert feedback led to the exclusion of non-critical instances due to their low practical relevance compared to the labelling effort required. As a result, the final dataset focuses on two semantically well-defined classes: *Critical*, denoting defects on or intersecting conductive wires that are likely to impact functionality, and *Dirt-Wire*, representing contaminants located on wires that warrant monitoring.

To ensure high data quality, a comprehensive relabelling pass was conducted, totalling approximately 45 hours of manual work. This effort involved tightening bounding boxes, removing ambiguous samples, and strictly enforcing class definitions, as we can see in Figure 1. The resulting improvement in label consistency yielded substantial performance gains, emphasising the value of data quality over architectural complexity in low-data regimes. The final dataset consists of 418 samples (131 *Critical* labels and 1054 *Dirt-Wire* labels), formatted in COCO [5] for compatibility with Slicing Aided Hyper Inference (SAHI), and divided into 80% for training and 20% for validation. A separate test set was omitted due to data scarcity.

IV. Methodology

The end-to-end experimental pipeline, from manual annotation to SAHI-enhanced inference, is illustrated in Figure 7 (Appendix).

A. YOLO-Based Detection Framework and Architectural Modifications

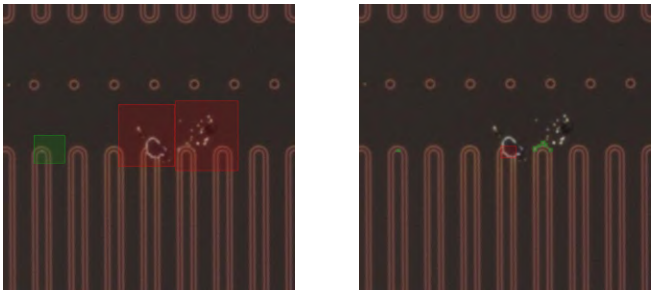
The proposed approach is based on the YOLO single-stage object detection framework [1], selected for its favourable trade-off between detection accuracy, inference speed, and model size. YOLO formulates object detection as a direct regression problem, predicting bounding box coordinates, objectness scores, and class probabilities in a single forward pass. Modern YOLO variants, such as the YOLO11 architecture utilised in this study, employ a fully convolutional backbone with multi-scale detection heads, enabling the detection of objects of varying sizes at different feature resolutions.

Standard YOLO11 architectures typically utilise three detection heads (P3, P4, and P5) with strides of 8, 16, and 32 pixels, respectively. While effective for general objects, these resolutions often prove insufficient for the extremely small defects (e.g., *Dirt-Wire*) present in microscope-acquired datasets. To address this, a modified **P2-head architecture** was implemented. This variant incorporates an additional detection layer with a stride of 4 pixels, preserving higher-resolution spatial information from the earlier stages of the backbone.

By integrating a P2-head, the model gains a specialised feature map for tiny objects, effectively doubling the resolution compared to the standard P3 head. Both the standard YOLO11 configuration and the modified P2-head variant were evaluated to isolate the impact of this architectural change on detection performance; the comparative analysis of these structural configurations is discussed in Section VI.

B. Data Augmentation and Class Balancing

To address the class imbalance problem, characterised by a significantly higher frequency of *Dirt-Wire* labels compared to *Critical* labels, two distinct resampling approaches were evaluated. The first involved *Oversampling* the critical defects combined with a targeted augmentation pipeline to



(a) Initial Labels (b) Refined Labels

Figure 1: The comparison highlights the transition from loose, inconsistent bounding boxes (left) to the tighter, standardized annotations (right). Green bounding boxes denote *Dirt-Wire*, while red indicates *Critical* defects.

Table I: Comparison of experimental configurations. The P2-head was evaluated against the baseline without augmentations to isolate its structural impact. Subsequent SAHI experiments incorporate data augmentations and compare different slicing resolutions and the inclusion of varying proportions of defect-free background images. Training time is reported in minutes (min) for the full 150-epoch duration. * denotes the best-performing configuration based on the highest Mean F1 score.

	Critical			**Dirt-Wire**			**Global Summary**			
Configuration	**P**	**R**	**F1**	**P**	**R**	**F1**	**mAP$_{50}$**	**mAP$_{50:95}$**	**F1$_{mean}$**	**Time (min)**
Initial Baselines										
Baseline (Full Image)	0.302	0.174	0.221	0.000	0.000	0.000	0.070	0.024	0.110	24.5
Baseline + Augmentations	0.003	0.130	0.005	0.000	0.000	0.000	0.012	0.001	0.003	34.4
Baseline + P2 Head	0.410	0.130	0.198	0.365	0.041	0.074	0.087	0.017	0.136	40.7
SAHI 128px (Oversampling + Aug)										
+ 50% Background	0.243	0.391	0.300	0.284	0.489	0.360	0.243	0.101	0.330	43.9
SAHI 256px (Oversampling + Aug)										
+ 10% Background	0.185	0.435	0.260	0.486	**0.571**	0.525	**0.368**	**0.152**	0.393	71.9
+ 30% Background	0.171	**0.522**	0.258	0.573	0.473	0.518	0.336	0.138	0.388	93.0
+ 50% Background*	0.435	0.435	**0.435**	0.674	0.478	**0.560**	0.323	0.126	**0.497**	155.1
+ 70% Background	0.412	0.304	0.350	**0.723**	0.401	0.516	0.238	0.092	0.433	208.4
SAHI 256px (Downsampling + Aug)										
+ 10% Background	0.217	0.435	0.290	0.352	0.462	0.399	0.303	0.130	0.345	11.5
+ 30% Background	0.320	0.348	0.333	0.396	0.302	0.343	0.164	0.069	0.338	14.1
+ 50% Background	0.364	0.348	0.356	0.497	0.434	0.463	0.236	0.104	0.409	19.2
+ 70% Background	**0.600**	0.261	0.364	0.516	0.363	0.426	0.231	0.098	0.395	31.2

mitigate overfitting, while the second utilised *Downsampling* of the majority class to align its distribution with that of the minority class.

The augmentation pipeline included Contrast Limited Adaptive Histogram Equalization (CLAHE) [6] to normalise contrast variations, synthetic sensor noise (Gaussian and ISO noise) to approximate microscope artifacts, and spatial degradations such as blur to simulate focus variations. These augmentations were implemented using the Albumentations library [7], selected for its computational efficiency in processing high-resolution image transformations within deep learning pipelines.

Furthermore, to mitigate false positives, the training pipeline was modified to integrate varying proportions of defect-free background images, a strategy known to improve the discriminative power of detectors in complex industrial backgrounds [8]. Both the resampling strategy and the background image ratio were treated as experimental variables. A systematic comparison of these configurations is presented in Section VI, identifying the optimal balance for final model deployment.

C. Slicing Aided Hyper Inference

To address the scale mismatch between image dimensions and defect sizes described earlier, Slicing Aided Hyper Inference (SAHI) [2] was integrated into the detection pipeline.

SAHI partitions each high-resolution image into overlapping tiles, which are processed independently by the detector. Predictions from all tiles are then projected back into the original image coordinate space and merged using non-maximum suppression. This inference-level adaptation preserves fine-grained spatial detail without requiring architectural changes. Hyperparameters were empirically tuned, with slice sizes increased from 128 px to 256 px to provide additional contextual information, overlap ratios set to 0.2 to avoid boundary artefacts, and the input resolution increased to imgsz $= 768$. The input resolution was chosen such that the smallest defects (4×4 pixels in a 256×256 slice) are enlarged to at least 12×12 pixels, exceeding the minimum effective detection size of the YOLO head; this was achieved by upscaling the input images by a factor of three. To ensure consistency, confidence and NMS IoU thresholds were set to 0.5 for all experiments.

V. Experimental Setup

All experiments were conducted locally on a single NVIDIA GeForce RTX 4060 GPU with 8 GB of VRAM; a summary of the experimental results is presented in Table I. Batch size, training duration, and model complexity were tailored to accommodate the available hardware resources.

Training was performed using a standard train-validation split, with evaluation conducted on a validation set. Due to computational constraints, cross-validation was not feasible; instead, model selection and hyperparameter tuning were guided by validation performance, with particular emphasis on recall given the high cost of false negatives. To ensure a fair comparison and simulate deployment conditions, SAHI was applied consistently across all slicing-based evaluations.

While recall was the optimization priority, final model selection was based on the F1-score to balance defect

coverage with an acceptable false-positive rate. This prevents excessive false alarms, which would otherwise increase downstream operational costs.

VI. Results and Discussion

To evaluate the impact of individual design choices, a structured ablation study was conducted in which only one major component of the pipeline was modified at a time. This incremental evaluation strategy allows observed performance differences to be attributed directly to specific methodological decisions.

The baseline configuration consists of a standard YOLO detector operating on full-resolution images without slicing or task-specific augmentations. Subsequent configurations introduce data augmentations, class balancing, YOLO P2 Head and inference-level slicing via SAHI with varying slice sizes. Table I summarises the quantitative performance across all evaluated configurations.

The results indicate that architectural complexity alone is insufficient for reliable defect detection in this setting. While data augmentation provides modest robustness gains, the most significant improvements are achieved through inference-level adaptations and improved annotation quality. In particular, the introduction of SAHI substantially increases recall by preserving fine-grained spatial information, confirming the importance of addressing extreme object–image scale mismatch. Increasing the slice size from 128 px to 256 px further improves precision by providing additional contextual information regarding superconducting circuit textures, effectively reducing spurious detections.

An intermediate architectural modification was also evaluated by introducing an additional P2 detection head to increase spatial resolution for small-object detection. This change enabled the baseline model to detect previously missed *Dirt-Wire* defects, yielding non-zero precision and recall for this class. However, training dynamics revealed that the model required a prolonged adaptation phase, with the mAP@0.50:0.95 metric remaining near zero for the first $\sim$60 epochs before gradually improving. While beneficial, this architectural adjustment increases the model training cost and does not fully address the extreme scale mismatch, motivating the transition toward inference-level adaptations via slicing.

Figure 2 shows the distribution of model predictions relative to the ground-truth labels. The results indicate that there is almost no confusion between the two defect classes. Compared with the confusion matrix produced by the baseline model (Figure 3), the improvement in detection performance is evident. This improvement is attributed to the applied data preprocessing, sliced inference, and class balancing techniques.

A. Fine-Tuning Strategy Analysis

To evaluate training efficiency under local compute constraints, different fine-tuning strategies were compared across YOLO model variants. The configurations included: *Default* (full fine-tuning), *All* (fine-tune only the last layer), and *Backbone* (freeze the backbone and train only the remaining layers). Figure 4 (located in the Appendix) shows plots of validation losses per epoch for the different freezing configurations. Representative results for the Nano version of the model (181 layers, 2.6M parameters, 6.6 GFLOPs) indicate that freezing backbone layers reduces training time by approximately 30% while achieving validation performance comparable to the default training configuration. Full fine-tuning consistently yielded the best final results but at a higher computational cost, suggesting that partial freezing is effective for rapid iteration, with full training reserved for final optimisation.

B. Qualitative Analysis and Practical Considerations

Qualitative inspection of detection outputs (see Figs. 5, 6 in the Appendix) confirms that the final model produces tighter and more consistent bounding boxes compared to earlier configurations, particularly for small defects located on conductive wires; however, for defects spanning only a few pixels, even minor localisation deviations can substantially reduce IoU-based mAP scores despite correct defect detection.

Beyond the core detection pipeline, we developed two auxiliary tools to support practical deployment. First, a lightweight graphical interface enables intuitive visualisation of model predictions. Second, an auto-labelling workflow integrates the trained detector with Label Studio, streamlining the annotation of new data. While these tools do not directly impact quantitative performance, they substantially improve annotation efficiency and facilitate iterative refinement of the dataset, making the overall workflow more practical and effective.

VII. Conclusion

This project demonstrates that reliable detection of critical defects in high-resolution microchip images is achievable under strict data and hardware constraints. Performance gains were driven by a data-centric strategy, combining rigorous relabelling and targeted augmentation with SAHI-based inference, which proved more effective than increasing architectural complexity alone.

The primary limitation is the small annotated dataset; the *Critical* category remains roughly two orders of magnitude below industry recommendations [8]. While hardware constraints limited experimentation with larger models, the results indicate that data-focused strategies are practical for industrial inspection. Future work should prioritize expanding the dataset and exploring semi-supervised learning techniques.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[2] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing aided hyper inference and fine-tuning for small object detection," *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 966–970, 2022.

[3] W. Liu *et al.*, "Small object detection in computer vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[4] M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov, "Label Studio: Data labeling software," 2020-2025. Open source software available from https://github.com/HumanSignal/label-studio.

[5] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll'a r, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[6] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*, pp. 474–485, Academic Press Professional, Inc., 1994.

[7] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," *Information*, vol. 11, no. 2, 2020.

[8] Ultralytics, "Tips for best training results." https://docs.ultralytics.com/yolov5/tutorials/tips_for_best_training_results/#dataset, 2025. Accessed: 2025-12-18.

Acknowledgements

We would like to thank the members of the Hybrid Quantum Circuits Laboratory (HQC) at EPFL, in particular Dr. Simone Frasca, Marius Mathias Bild, Hon Ming Andrew Yip and Antoine Clément Silvin, for providing domain expertise, access to microscopy data, and valuable feedback throughout the project. We also thank the teaching staff of the CS-433 Machine Learning course for their guidance and support.

VIII. Appendix

Ethical Risks

The primary ethical risk identified in this project is the occurrence of *false negatives*, where critical defects in microchip images are not detected. The stakeholders affected include manufacturing engineers and quality control teams who may rely on the model during inspection, as well as downstream users who are indirectly impacted if defective chips pass quality control. Undetected defects can lead to functional failures, economic losses, and reduced trust in automated inspection processes. Given the extremely small size of defects and the limited availability of labelled data, this risk is non-negligible in both severity and likelihood.

The risk was evaluated through a quantitative analysis of model performance, with particular emphasis on class-specific recall, confusion matrices, and precision–recall trade-offs. Early baseline experiments exhibited zero recall for certain defect classes, highlighting the inadequacy of standard detection pipelines and motivating subsequent architectural, data-centric, and inference-level refinements. Recall was therefore monitored as a key indicator of progress, while overall performance was assessed using the F1-score to balance defect coverage with false-positive rates. Mitigation strategies included slicing-based inference to preserve fine-grained spatial information and the incorporation of defect-free background images to improve discrimination. The system is explicitly designed as a decision-support tool rather than an autonomous decision-maker, with final judgments remaining under human control.

No ethical risks related to privacy or data protection were identified. The dataset consists exclusively of sensitive but non-personal microchip imagery, with access restricted to the project team. All annotation, training, and evaluation were performed locally, without the use of cloud services or external computing resources, ensuring that proprietary data was not transmitted or exposed.

Results Data

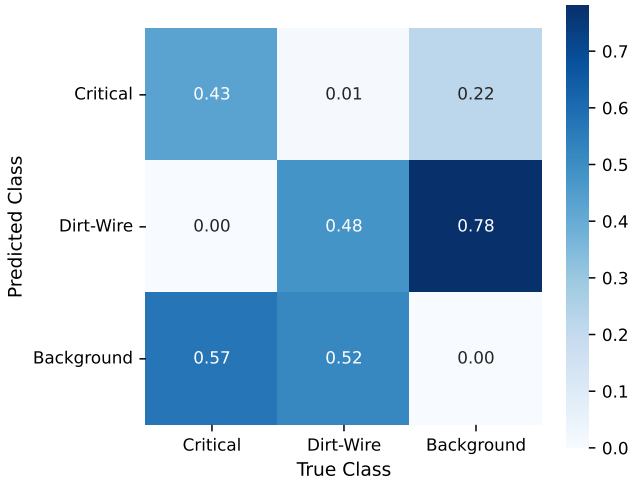


Figure 2: Normalized confusion matrix for the final model. Predictions are evaluated at a confidence threshold of 0.5.

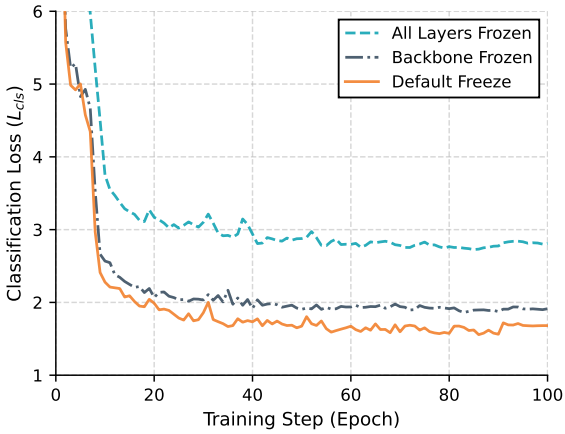


Figure 4: Validation classification loss of the baseline nanoYOLO model under various layer-freezing strategies during fine-tuning. Training times for 100 epochs: All Layers Frozen (16.1 min), Backbone Frozen (14.4 min), and Default/Full Fine-tuning (20.1 min).

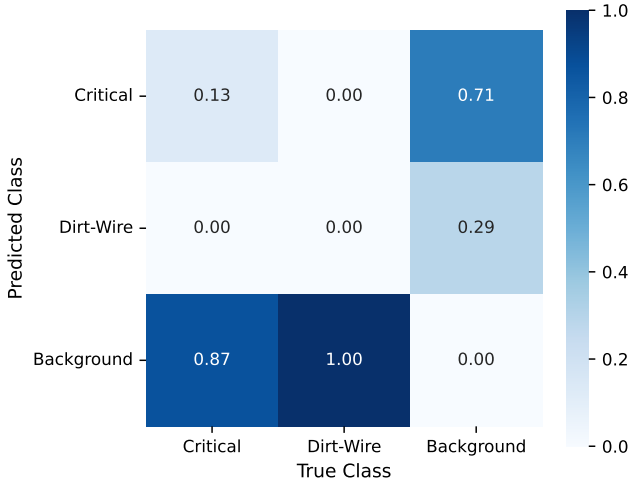


Figure 3: Normalized confusion matrix for the baseline model. Predictions are evaluated at the optimal confidence threshold of 0.179.

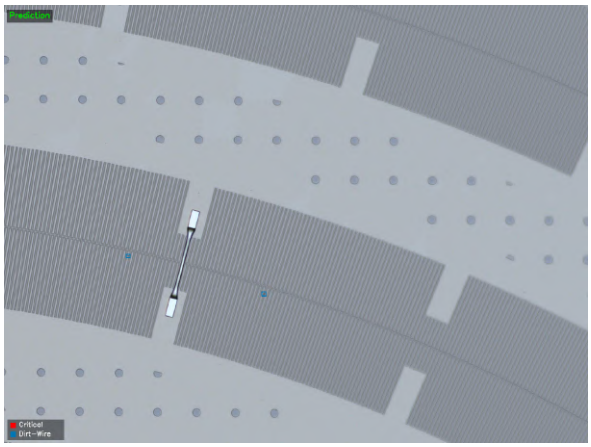


Figure 5: Example of a processed input image showing a portion of a superconducting chip with detected defects highlighted by bounding boxes.

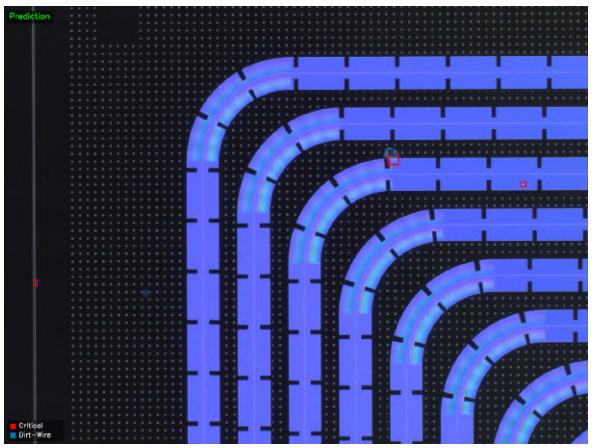


Figure 6: Example of a processed input image showing a portion of a superconducting chip with detected defects highlighted by bounding boxes.

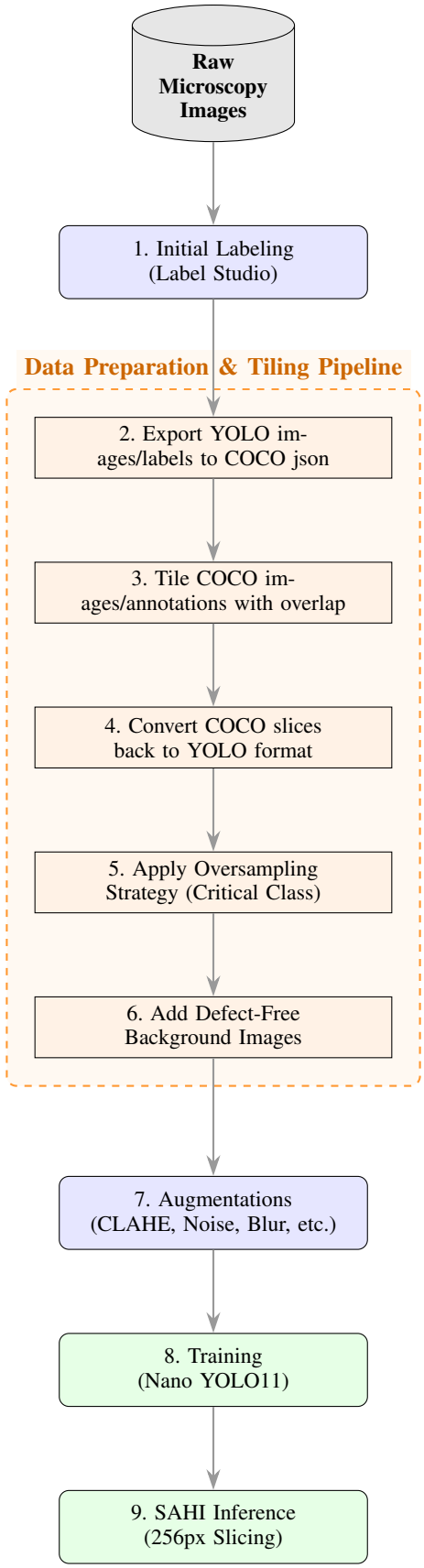


Figure 7: Complete project workflow illustrating the transition from raw data through the specialized tiling and preprocessing pipeline, leading to model training and Slicing Aided Hyper Inference (SAHI).