

Article

Not peer-reviewed version

In-Situ Screening of Defective Electronic Components through Edge Real-Time Big Data AI Analysis

[Eyal Weiss](#)*, Shir Caplan, Jacob Horn, Moshe Sharabi

Posted Date: 15 March 2024

doi: 10.20944/preprints202403.0870.v1

Keywords: real-time; image processing; quality; manufacturing; defect detection; inspection; TensorFlow; kubernetes; gRPC; production optimization



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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

Article

In-Situ Screening of Defective Electronic Components through Edge Real-Time Big Data AI Analysis

Eyal Weiss *, Shir Caplan, Kobi Horn and Moshe Sharabi

Technology Department, Cybord.ai, Tel-Aviv, Israel

* Correspondence: eyal.w@cybord.ai

Abstract: This paper presents a groundbreaking approach to real-time image processing in electronic component assembly, enhancing quality control in manufacturing. By capturing images from pick and place machines between component pickup and mounting, defects are identified and addressed in-line, significantly reducing the likelihood of defective products. Leveraging fast network protocols like gRPC and orchestration with Kubernetes, along with C++ programming and TensorFlow, this method achieves an average turnaround time of less than 5 milliseconds. Tested on 20 live production machines, it ensures compliance with IPC-A-610, and IPC-STD-J-001 standards while optimizing production efficiency and reliability.

Keywords: real-time; image processing; quality; manufacturing; defect detection; inspection; TensorFlow; kubernetes; gRPC; production optimization

1. Introduction

The electronic manufacturing industry confronts a formidable challenge stemming from the high incidence of failures, largely attributed to the pervasive assumption regarding the inherent reliability of utilized components during the assembly process [1–6]. Manufacturers traditionally concentrate their efforts on refining the assembly process itself, often overlooking the critical necessity for rigorous materials inspection. This conventional approach entails procuring components from trusted sources [7–10].

These risks encompass a broad spectrum of issues, notably a lucrative market for fraudulent components valued at approximately \$70 billion annually [11–17]. Fraudulent components manifest in diverse forms, ranging from counterfeits and fakes to substandard and disqualified products, exacerbating the complexity of the challenge. Moreover, inadvertent errors introduced by operators within the materials supply chain and during the production process further compound the issue, resulting in the inadvertent utilization of incorrect materials [12,18,19]. Furthermore, even when components are sourced from reputable suppliers, the potential for encountering random defects within a batch remains a pressing concern, with an average occurrence of 100-200 defects per million (DPM) (see examples in Figure 1 and Figure 2) [20–24].



Figure 1. Examples on Passive Components (Left to Right): Conductive FOD (Foreign Object Debris), Corrosion leading to cracks, Leads peeling, Non-conductive FOD.

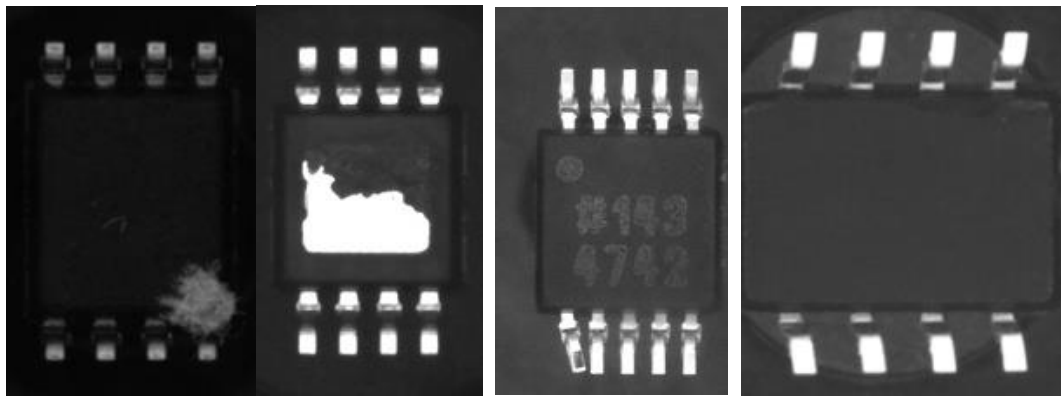


Figure 2. Examples on Lead Components (Left to Right): FOD, Termination peeling, Bent leads, Distorted leads.

These defects, albeit challenging to predict and manage, can significantly undermine the reliability of the final product.

What constitutes a body defect in an electronic component? A body defect, though often subtle and barely discernible to the naked eye, can wield significant ramifications in today's demanding applications characterized by stringent standards and heightened stress levels [20,22–24]. Even minor imperfections, whether in a cheap capacitor or a more expensive processing unit, harbor the potential to trigger product malfunction. It can be disheartening to witness the failure of a top-tier product due to an unnoticed defect in a seemingly inconspicuous component valued at just one cent. Body defects encompass a spectrum of issues, including cracks, breaks, peeling of metallization, deformations, discoloration, mold, corrosion, bent leads, deformed leads, and misshapen BGA balls, among others [25]. Illustrative examples are presented in Figure 1 and Figure 2. IPC-A-610H and IPC-STD-J-001 as the leading standards in electronic manufacturing present a comprehensive definition of defects on the assembled PCB's. A detailed description is presented in this work.

Compounding these challenges are instances where components are inadequately handled or stored, exposing them to environmental factors such as humidity, acidity, and other contaminants. Additionally, components may sustain damage from vibration and shock during transit, leading to latent defects that may only manifest in the final product [1,2,26].

Conventional production processes have traditionally operated under the assumption of flawless component quality, without integrating technologies for comprehensive component testing [3,27]. The sheer magnitude of electronic component usage, encompassing trillions of components annually across myriad component types, renders conventional incoming inspection impractical. Consequently, the industry has reluctantly accepted the economic burden of substandard materials.

However, innovative technologies are heralding a potential solution to this entrenched issue [20,22–25]. The method proposed in this paper capitalizes on existing component cameras integrated into pick and place machines, capturing images of each component immediately before mounting it on the PCB, from the component's underside. The presented method harnesses these images, deploying a sophisticated AI algorithm to authenticate components, detect defects, and identify corrosion [7,25].

Nevertheless, a fundamental constraint of the prevailing system resides in cloud-based analysis, necessitating several seconds to process the data. Consequently, alerts regarding component issues are issued after the component is already on the PCB, precipitating entire pallet rejection and subsequent rework. In sectors such as automotive, medical, and aerospace, where rework is impermissible, the entire board may be scrapped, precipitating significant material and financial losses, alongside contributing to electronic waste (e-waste) [28–32].

The key innovation advanced in this paper lies in the capability to process data in real-time, facilitating the detection and rejection of defective components between their pick-up and placement on the PCB. This advancement ensures that defective components are preemptively rejected,

mitigating the risk of assembly failures, reducing waste, and augmenting the reliability of electronic products.

In the domain of electronic component assembly, real-time inspection assumes paramount importance owing to the rapid and dynamic nature of modern pick and place machines. These machines demonstrate remarkable efficiency, mounting components at rates exceeding 100,000 components per hour. Consequently, a mere 36 milliseconds, on average, is available for placing each component, and in some cases the detection alarming window is as short as 10 milliseconds, underscoring the exigency for real-time inspection amidst this frenetic operational tempo.

As we go into the exploration of the presented AI-driven inspection method, it's important to emphasize the dedication of the industry to standards. Among the pivotal standards in electronic component inspection lies the IPC-A-610H, and IPC-J-STD-001 standard, a widely acknowledged benchmark for defining defect criteria and ensuring assembly quality. In the upcoming section, we examine how the presented method not only aligns with but also elevates compliance with IPC-A-610H and IPC-J-STD-001 standards, thereby ensuring defect detection and adherence to industry best practices [33].

Deep learning algorithms excel at analyzing vast amounts of image data with precision, enabling the detection of subtle defects that may elude human perception. These algorithms can discern anomalies ranging from surface imperfections to structural irregularities, thereby facilitating early detection and mitigation of potential issues. Moreover, the real-time processing capabilities of deep learning systems are instrumental in ensuring timely defect identification and rejection during the assembly process. By analyzing component images captured by pick and place machines instantaneously, these systems can preemptively flag defective components before they are mounted onto the PCB, thereby minimizing the risk of downstream failures. Furthermore, the adaptive nature of deep learning models enables continuous improvement and refinement over time, ensuring robust performance even in the face of evolving challenges and variations in component characteristics. This adaptability enhances the overall resilience and effectiveness of the inspection system, contributing to sustained improvements in product quality and reliability.

2.1. Ensuring Compliance with IPC-A-610H and IPC-J-STD-001 Standards

The IPC-A-610H standard serves as a comprehensive guide for assembled PCB's, outlining specific criteria for identifying defects and ensuring the quality of electronic assemblies. Other standards relate to the specific defect details as they appear in the IPC-A-610. In this section, we integrate the key compliance parameters outlined in IPC-A-610 with the presented AI-driven inspection system. In each section we highlight the standard sections that are relevant to the presented method and that the method can detect automatically on all inspected components. The detection algorithm is only mentioned in this work and is elaborated in [20,22–25]. In this work the novelty is in adapting the methods to real-time operation.

Defects on Component Leads/Terminations - Section 8.2.2

The examination of component leads is integral to our defect detection process, aligning with IPC-A-610's criteria. Leveraging deep learning algorithms, the presented method scrutinizes each lead for damage or deformation exceeding 10% of the lead's diameter, width, or thickness, ensuring compliance with IPC-A-610 standards. Examples from images taken by the pick and place machines and disqualified by the algorithm are presented in Figure 3. The deviations are circled. The visible defects are scratches, dents, distortions, deformations, peeling, shorts, etc.

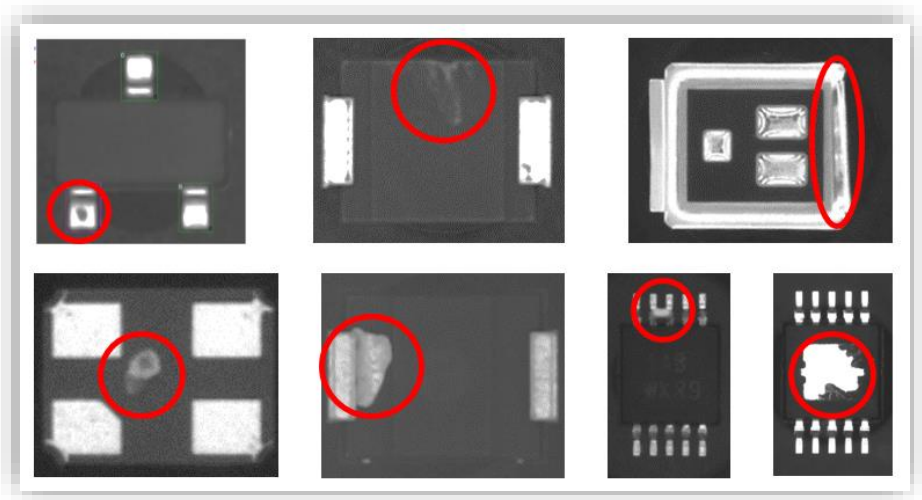


Figure 3. Examples from images taken by the pick and place machines and disqualified by the algorithm.

Bent or Warped Leads - Section 8.3.5.8.

The presented method, powered by advanced deep learning techniques, is adept at identifying bending, indentation, and coplanarity issues in component leads, in accordance with IPC-A-610 Section 8.3.5.8. Through the presented advanced evaluation techniques, we ensure that any deviation beyond the specified threshold is promptly flagged for further assessment. Figure 4 presents examples of bent leads and coplanarity issues automatically detected by the algorithm that exceeds 10% of the lead’s width.

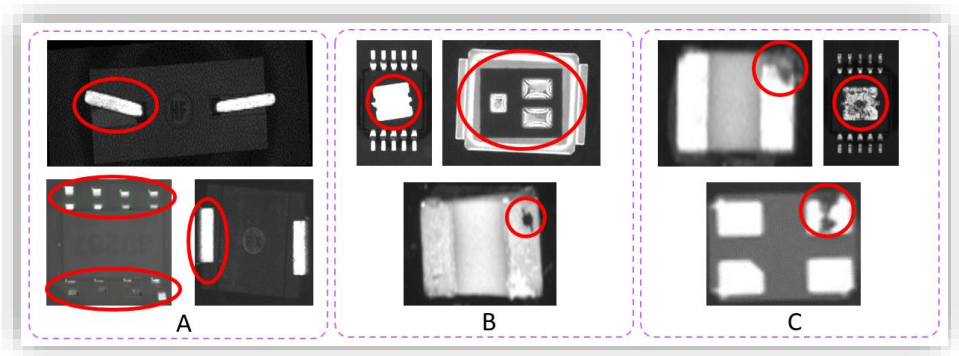


Figure 4. Examples of bent leads and coplanarity issues automatically detected by the algorithm that exceeds 10% of the lead’s width. (A) bent leads, (B) deformed leads, (C) damaged leads.

Corrosion and Cleanliness - Section 10.6.4

The presented method is equipped to detect corrosion and residues on metallic surfaces with precision. By promptly recognizing any indication of discoloration or evidence of corrosion, it ensures compliance with IPC-A-610 cleanliness and surface appearance parameters. Figure 5 presents examples of components with corrosion and contamination automatically detected by the AI algorithm that exceed 10% of the lead’s width.

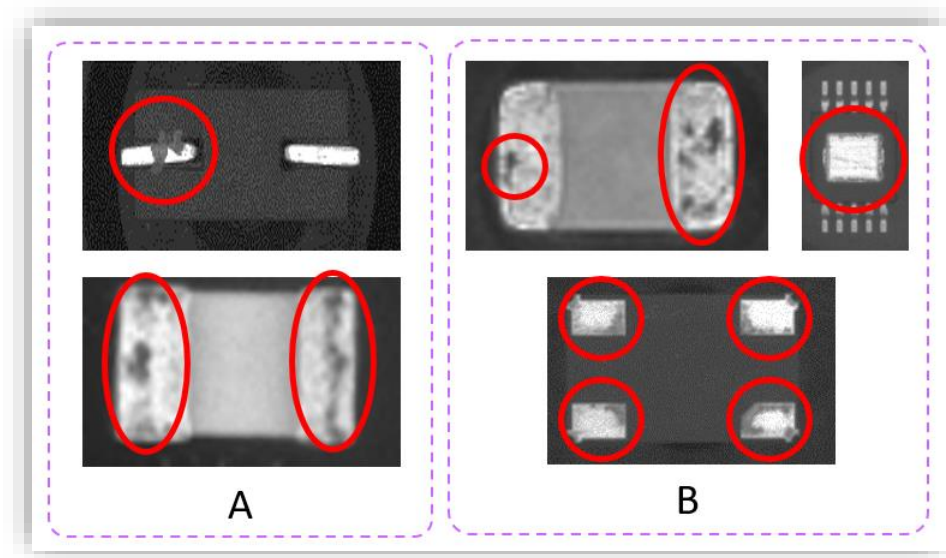


Figure 5. Examples of components with corrosion and contamination automatically detected by the AI algorithm that exceed 10% of the lead's width. (A) Contamination of leads, (B) Corrosion of leads.

Cleanliness - Foreign Object Debris (FOD) - Sections 10.6.2 & 10.6.3

IPC-A-610 emphasizes the importance of cleanliness in electronic assemblies, particularly concerning foreign object debris (FOD). The presented method evaluates components for contamination, flagging any debris or residues beyond the specified threshold for further evaluation, in line with IPC-A-610 Section 10.6.2 and 10.6.3 standards. Figure 6 presents examples of components with corrosion and contamination automatically detected by The AI algorithm.

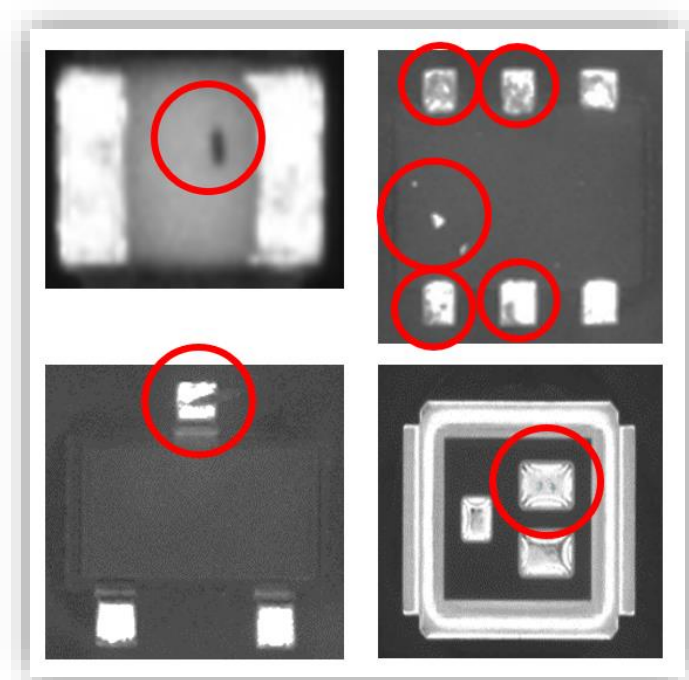


Figure 6. Examples of components with corrosion and contamination automatically detected by The AI algorithm. Note that debris may be due to a component defect or point out a root cause in the supply chain, such as contaminated components in a container or poorly handled material.

Loss of Metallization - Section 9.1, 9.3

Metallization loss defects are critical vulnerabilities highlighted by IPC-A-610 standards. The presented method identifies irregularities in metallization coverage, ensuring optimal functionality and reliability of electronic components as per IPC-A-610 Section 9.1 and 9.3 standards. Figure 7 presents examples of components with metallization delamination automatically detected by the AI algorithm.

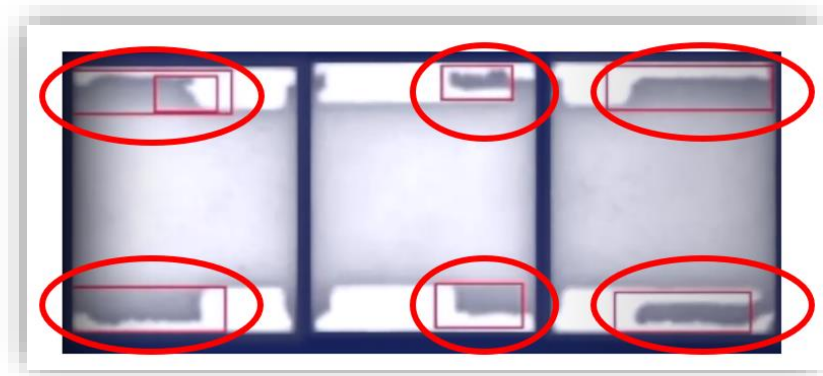


Figure 7. Examples of components with metallization delamination automatically detected by the AI algorithm.

Mounting Upside Down - Section 8.3.2.9.2

Lastly, the presented method is designed to detect components mounted upside down, as specified in IPC-A-610 Section 8.3.2.9.2. By flagging any non-compliant mounting configurations, our system ensures adherence to IPC-A-610 standards. Figure 8 presents A resistor mounted upside-down from the top view.

In summary, the integration of advanced deep learning algorithms enhances the inspection system seamlessly integrates IPC-A-610 compliance parameters, providing a robust framework for ensuring the quality and reliability of electronic components in accordance with industry standards.

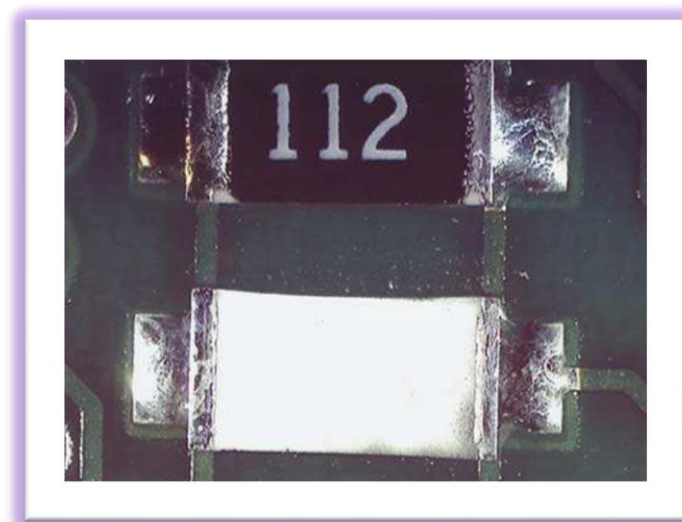


Figure 8. A resistor mounted upside-down from the top view.

2. Design

2.1. Detection of Defects and Corrosion Issues by Deep Neural Network

Detecting defects and corrosion in electronic components is a crucial task within the manufacturing process, presenting a challenge in image classification where the objective is to categorize images as either "normal" or "defective." In some instances, defects and corrosion may warrant finer categorizations, such as minor, moderate, or major. Initially, efforts were made to leverage popular ImageNet models to tackle this classification challenge; however, these models proved to be inadequate due to two principal challenges:

Limited Training Data: One of the primary challenges arises from the scarcity of training data available in the context of electronic components. The proportion of defective items within electronic components is minuscule, often amounting to less than one defect per ten-thousand components [24]. Gathering a sufficient number of defective items to establish a robust training dataset proves to be a daunting task. In scenarios where only a few hundred defective items are accessible among millions of components, conventional image classification models tend to suffer from overfitting. Overfitting leads to a situation where training accuracy reaches 100%, but validation accuracy remains considerably lower. In some cases, the validation accuracy may not surpass that of random guessing.

Similarity of Defective Items: The second challenge revolves around the striking similarity between some defective items and their normal counterparts. Defects and corrosion can manifest in subtle forms, such as small dots within the components. Conventional Convolutional Neural Network (CNN) models often struggle to effectively identify these inconspicuous defects and corrosion due to their limited capacity to capture fine-grained details [22].

To address these challenges effectively, we introduce a network architecture specially tailored for defect and corrosion detection presented at [20,21] (see Figure 9). In contrast to the serial architecture commonly found in popular image classification models, where multiple layers of small filters, typically 3x3 in size, are connected sequentially, our approach adopts a set of filters with varying window sizes that operate in parallel. Each filter within this set is specifically designed to detect a particular type of defective region or corrosion within component images. These filters generate features that indicate the likelihood of such defects and corrosion being present. The overall feature vector is formed by concatenating all the individual features generated by these parallel filters. Subsequently, this feature vector is passed through a dense layer, which produces the final defect and corrosion score (see Figure 9).

In addition, we use data augmentation techniques during training to enhance the network performance. These techniques involve artificially expanding the training dataset by applying transformations such as rotation, flipping, and scaling to the existing images. By generating variations of the original data, augmentation helps prevent overfitting and improves the model's ability to generalize to unseen data. Thus, augmentation is enabling them to learn from a more diverse range of examples.

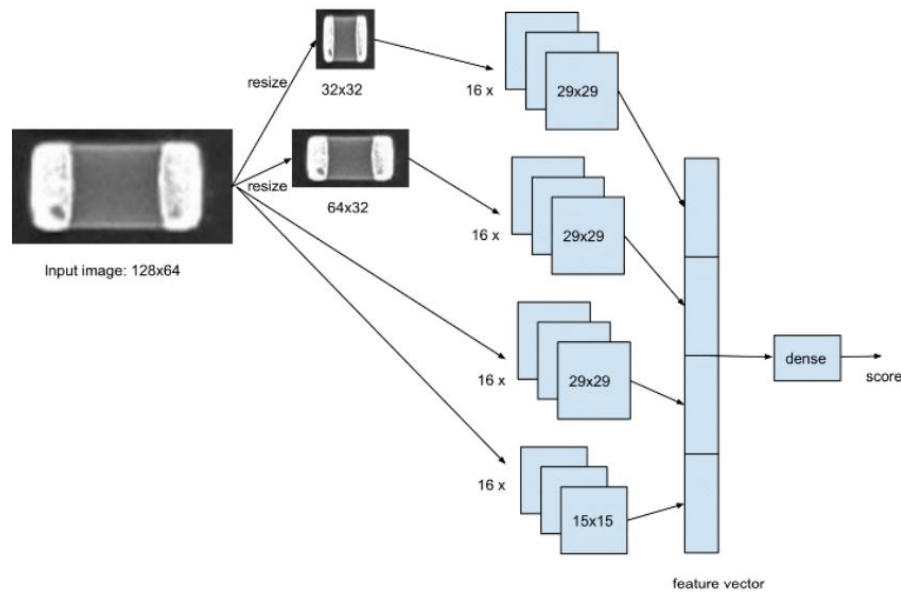


Figure 9. Neural network architecture [22–24].

Speeding Up with GPU Acceleration: The utilization of this parallel architecture not only enhances the accuracy of defect and corrosion detection but also significantly improves the efficiency of the algorithm. It allows for high-speed processing, with the capability to analyze up to 3,000 images per second on a Tesla T4 GPU. This GPU acceleration results in a substantial reduction in processing time, making the system well-suited for real-time inspection during the component assembly process. By optimizing the architecture for GPU support, the algorithm harnesses the parallel processing capabilities of the GPU to perform image analysis rapidly. The combined advantages of this architecture and GPU acceleration ensure efficient and timely defect and corrosion detection, further enhancing the overall quality of electronic products.

2.2. Operational Flow and Intervention Points in Pick-and-Place Component Handling

Figure 10 provides an overview of the operational flow, illustrating the journey from image capture to component placement. This visual representation underscores critical intervention points and the execution of commands for component rejection. Highlighting the process window emphasizes the role of timely actions in upholding assembly process quality and reliability. Notably, the response time window ranges from 10 to 30 milliseconds, contingent upon the machine model.

The component shooters, renowned for their swiftness, are picking up 20 to 28 components in a single run. They comprise two main families: the revolver pick-up head, typical of ASMPT, adept at picking up components via a revolving head, and the cluster-based pick-up head typical to Fuji, and Yamaha with multiple nozzles, sequentially picking up components before undergoing translation across a pit camera. Although rapid, shooters are primarily suited for small components, resulting in relatively small image sizes and short processing times.

Conversely, pick-and-place heads, tailored for larger components, pick up a smaller quantity per run, albeit requiring more time for processing due to the larger component size. These contrasting parameters strike a balance, leading to comparable speed requirements across all pick-up heads.

The role of the pick and place machine lies in capturing component images to ascertain their position on the pick-up nozzle, enabling precise placement despite minor position variations. Additionally, the algorithm identifies soldering lead locations and detects faulty pick-ups, such as empty nozzles or tombstone pick-ups. Conventional machine

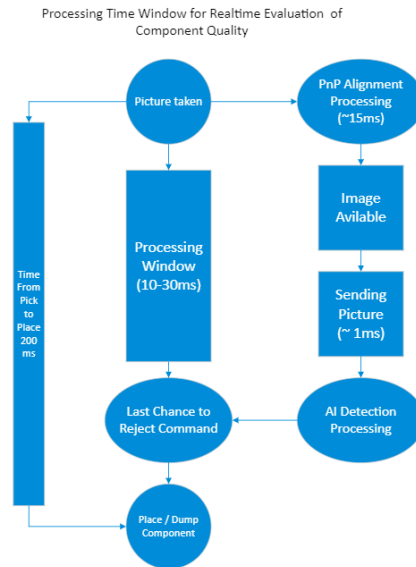


Figure 10. Process flow of the pick and place machine illustrating the timeline from image capture to component placement, emphasizing the available process window for intervention and commanding component rejection.

vision tools process these images swiftly between pick-up and placement. Any deviation from acceptable pick-up thresholds prompts the vision software to flag the component for rejection, subsequently leading to its disposal in a designated bin.

The presented method mirrors this rejection mechanism and aligns with the time frame for image processing outlined in Figure 10, ensuring swift identification and rejection of faulty components.

2.2. Data Collection from the Pick and Place Machine

Data collection from the pick and place machine is streamlined through the deployment of a dedicated Application Programming Interface (API), purpose-built to facilitate the transfer of images and metadata from the pick and place machine to a local server.

Custom API for Image Transfer: The data collection process is based on an API deployed into the pick and place machine. This Rest-API interfaces with the machine's image capture system, orchestrating the collection of data in a format conducive to real-time analysis. Its primary role encompasses retrieving images of components milliseconds before their placement on the PCB, capturing visual information for defect and corrosion assessment. Moreover, the API gathers metadata linked with each image, furnishing context for subsequent analysis.

GRPC Protocol for Efficient Data Transmission: The data amassed by the API undergoes transmission to a local server utilizing the Google Remote Procedure Call (GRPC) protocol. GRPC is an optimal choice for this application. It facilitates swift and reliable transmission of images and metadata from the pick and place machine to the local server. This protocol's adeptness in real-time applications ensures low latency and high throughput, pivotal for timely defect and corrosion detection. Data exchange between the pick and place machine and the local server transpires within the regular local network.

2.3. Pre-Processing, Feature Extraction, and Model Architecture

The effectiveness of the presented real-time defect and corrosion detection system relies on a series of procedures, encompassing pre-processing, feature extraction, and the selection of an appropriate model architecture. The following sections delineate each of these steps in meticulous detail.

Pre-processing for Image Enhancement

Centering and Cropping: The pre-processing workflow commences with centering and cropping of the component image. This initial step involves aligning the component centrally within the image frame and cropping it to maintain a uniform background, thereby facilitating consistent and accurate analysis. Figure 11 presents a series of sample cases illustrating instances that the algorithm must disregard due to the pick-and-place machine's vision system's capability to detect them. Additionally, it showcases an example of a component image that is not centered, highlighting the challenges in processing such images accurately.

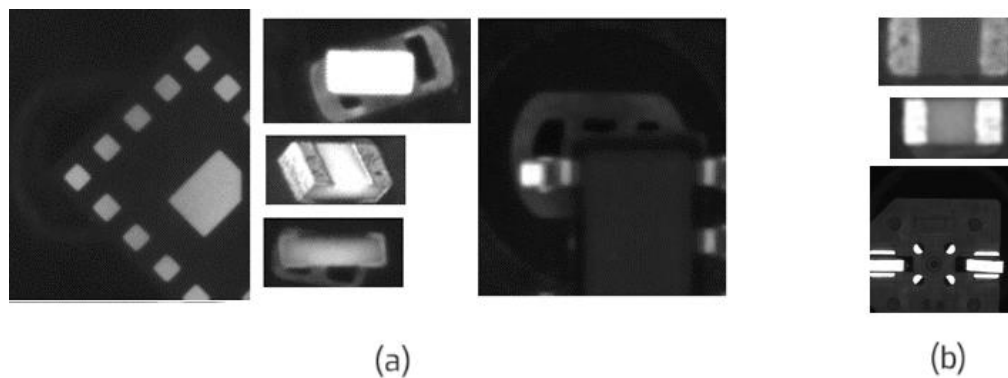


Figure 11. Sample Cases of Pick-Up Quality and Component Alignment. (a) Sample cases demonstrating pick-up instances that the algorithm should ignore, as the pick-and-place machine's vision system is capable of detecting and addressing these issues. (b) Example of a component image where the component is not centered, illustrating the challenges in processing such images accurately.

Light Balance Correction: is a step in the pre-processing phase, aimed at addressing any lighting irregularities within the images. If the image is not well balanced the algorithm will not be able to see subtle deviations in the picture. A dedicated classifier is utilized to detect images with lighting discrepancies, specifically those suffering from saturation. Upon identification, the system promptly notifies the machine to make necessary adjustments. Correcting these imbalances is vital as saturated images not only hinder analysis but also compromise the accuracy of component placement, potentially increasing the attrition rate. Figure 12 illustrates an example of a component with excessive light balance, highlighting the significance of this correction process.

Blurry Image Detection: In addition to addressing light balance issues, the pre-processing stage involves identifying and handling blurry images. Blurry images can significantly impact the accuracy of component placement and inspection. These blurs often occur due to incorrect programming, such as insufficient resolution settings (binning) or incorrect component height programming, leading to poor focus images. Additionally, defective lighting or imaging systems can also contribute to image blurriness. Detecting and addressing blurry images is crucial to ensure the reliability and accuracy of subsequent analysis steps. Figure 13 illustrates an example of a blurry image captured during the component inspection process, highlighting the importance of detecting and mitigating such issues during pre-processing.



Figure 12. Example of a component with excessive light balance, illustrating the importance of light balance detection in pre-processing.

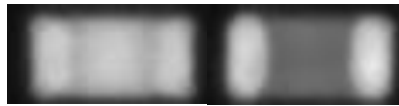


Figure 13. Example of a blurry image captured during component inspection, emphasizing the need for effective detection and correction of blurry images in the pre-processing stage.

Feature Extraction for Defect and Corrosion Identification

Feature extraction is discerning normal components from those exhibiting defects or corrosion. The presented approach adopts a distinctive feature extraction methodology to surmount the challenges associated with limited training data and subtle defects.

Metadata Matching: In the presented methodology, metadata integration plays a role in refining the analysis process. By correlating image metadata with the actual image data, we glean insights into the component's package type, facilitating the selection of the most appropriate model for analysis. This strategic alignment ensures that the algorithm is calibrated to address the nuances and potential defects or corrosion specific to each component package.

Package Type Insight: A key aspect of metadata matching involves gaining understanding of the component's package type. This insight enables us to categorize components effectively based on their visual similarities. For instance, in the case of passive chip components, such as chip resistors, classification is determined by their functionality and footprint. For example, a chip resistor with a size of 0603 is denoted as CRES-0603. It's important to note that the algorithm does not necessitate the specific resistor value or part number for analysis. Instead, our approach focuses on identifying and categorizing components based on their production lines and visual characteristics [21].

Model Architecture for Defect and Corrosion Detection.

3. Advancements to Real-Time Processing

In the Traditional Approach data collection involved capturing component images and associated metadata, which were subsequently transmitted to cloud servers for processing [16,21–25,34]. While this cloud-based analysis offered comprehensive analysis of each component, it was hampered by a significant drawback. The processing time per component was notably protracted, averaging around 10 seconds. In earlier iterations, processing occurred in batches of similar components, leading to delays in defect and corrosion detection. This latency primarily stemmed from the time required for data transfer to and from the cloud.

In the presented method outlined in this study prioritizes real-time defect and corrosion detection. A crucial aspect of this innovation involves relocating the processing center. Instead of relying on cloud-based analysis, the presented approach emphasis on edge computing, ensuring that processing transpires directly at the point of data acquisition. By conducting the analysis on-site, within the brief interval between image capture and component placement, we obtain the detection and resolution of defective components before they are placed on the PCB. To accommodate the velocity required for real-time defect and corrosion detection, the novel method accentuates the utilization of Graphics Processing Units (GPUs). The refinement of algorithms and software capable of operating within the stringent timeframes allotted for analysis.

The requirement of achieving real-time processing for Deep Neural Network (DNN) classification models on edge devices, resembling MobileNet, necessitates the exploration of techniques and tools to boost processing speed. This section goes into methods, elucidating their functioning, evaluating their merits and drawbacks, and recommending a suitable library or tool for our specific application.

Quantization involves reducing the precision of model weights and activations, typically transitioning from 32-bit floating-point numbers to 8-bit fixed-point numbers [35,36]. This reduction curtails memory usage and computational demands, facilitating swifter inference. Quantization is advantageous for its efficacy in diminishing the model's memory footprint and expediting computations, particularly beneficial for resource-constrained edge devices. Nevertheless, it may lead to a reduction in model accuracy due to diminished precision.

Model pruning entails eliminating non-essential weights or neurons from the neural network, resulting in a more streamlined and expedited model [37]. Pruning can occur during or after training. Pruning is beneficial as it significantly reduces the model's size, leading to quicker inference times and a smaller memory footprint. However, it necessitates meticulous handling to avert accuracy loss and may introduce some model instability. The TensorFlow Model Optimization Toolkit furnishes the requisite tools for model pruning and post-training quantization, rendering it a pragmatic choice. Depthwise Separable Convolutions (DSC) split the convolution process into two stages: depthwise convolution applies a filter to each input channel independently, followed by pointwise convolution to combine the results. This approach reduces computational cost and is commonly used in architectures like MobileNet for efficient inference on mobile and edge devices.

Model parallelism divides the model into smaller segments, each processed in parallel on distinct GPUs. This approach can markedly hasten inference on systems endowed with multiple GPUs [38]. Model parallelism is advantageous for its potential to achieve linear speedup with additional GPUs, particularly valuable for intricate models with high computational demands. Nevertheless, it mandates multiple GPUs and may not be suitable for edge devices equipped with a sole GPU. Both PyTorch and TensorFlow extend support for model parallelism on GPUs.

Operator fusion amalgamates multiple mathematical operations into a single operation, mitigating overheads associated with memory transfers and kernel launches on GPUs. Operator fusion's benefits encompass augmented inference speed by curtailing the number of kernel launches, often a bottleneck on GPUs. Nevertheless, the implementation may be intricate and necessitate low-level code modifications. Various deep learning frameworks, including TensorFlow and PyTorch, facilitate operator fusion optimizations.

The ONNX Runtime serves as an inference engine optimized for executing ONNX models, furnishing high-performance inferencing across diverse hardware platforms, including GPUs. ONNX Runtime proves advantageous for its efficiency in model execution, rendering it an excellent choice for accelerating inference on GPUs. Available as a standalone library, ONNX Runtime is compatible with various deep learning frameworks supporting the ONNX format.

In the deployment of MobileNet-like models on GPUs at the edge, it is advisable to consider quantization and model pruning. These methods substantially reduce model size and enhance inference speed while maintaining an acceptable level of accuracy. TensorFlow, with its comprehensive support for quantization and model optimization, emerges as a practical platform for implementing these optimizations. Additionally, ONNX Runtime can be leveraged for efficient inference on edge devices. The discerning utilization of these methods empowers the attainment of real-time inference on the edge, fostering rapid and precise execution of MobileNet-like models for electronic component inspection.

4. Architectural Framework and Deployment Strategies

This section goes into the architectural framework of the solution and the strategies deployed to optimize its performance and reliability. The architecture is presented in Figure 14. The effectiveness of this architecture is underscored by its demonstrated performance, which is elaborated in table I. This approach constitutes a method to address the challenges posed by limited training data and the identification of subtle defects and

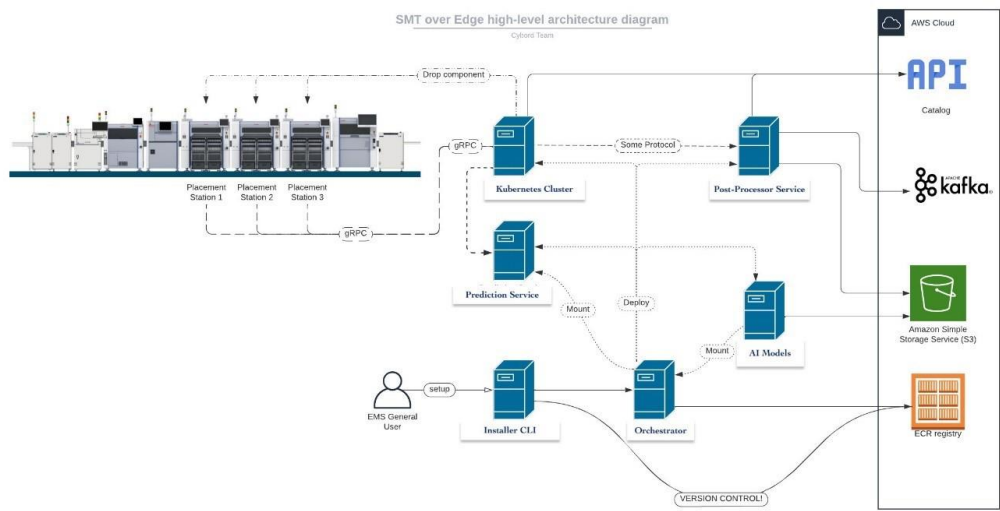


Figure 14. SMT over edge high-level architecture diagram.

corrosion in electronic components during the assembly process. It offers an efficient, data-efficient solution for the detection of defects and corrosion, ultimately contributing to the enhancement of product quality and reliability.

The solution harnesses Kubernetes to orchestrate services across multiple servers at each customer's site, enhancing system robustness and scalability. Machine learning models are stored in cloud storage, facilitating on-demand retrieval during software startup. These models are securely encrypted in memory to ensure controlled access. An external process continuously monitors changes in remote storage, directing the system to fetch updates, thus ensuring the deployment of the most current models. The system operates on Dell servers equipped with Nvidia RTX GPU. Adopting a Software as a Service (SaaS) model, the system receives data through the gRPC protocol, while model execution is executed using C++ TensorFlow Serving. The architectural design processes asynchronous data from all machines at the site, enabling concurrent execution of two AI models for each component. Commands are issued to deactivate components only when necessary. Figure 15 shows the Kubernetes cluster, based on Nvidia's GPU, scaling to accommodate multiple production machines and growing production demands.



Figure 15. The graphs show the Kubernetes cluster, based on Nvidia's GPU, scaling to accommodate multiple production machines and growing production demands.

5. Experimental Setup and Evaluation

This section describes the experimental setup employed to evaluate the solution's performance and capabilities. The experiments encompassed a range of hardware configurations and software optimizations aimed at evaluating its potential in real-world electronic component assembly scenarios. We utilized the Nvidia RTX 3070 server, featuring 8GB of DDR6 RAM and 16 cores, as the foundational server for our tests. To simulate real-world conditions, we first replicated an environment mirroring production line setup, comprising 20 Pick and Place (P&P) machines continuously dispatching components via the gRPC protocol. Our experimentation extended to exploring Nvidia's A100 and L4 GPUs to evaluate adaptability to different GPU configurations. C++ served as our primary programming language. We conducted comparative assessments to discern performance differences between versions. Additionally, we employed TensorRT, a deep learning inference optimizer and runtime library developed by NVIDIA. Integration of TensorRT enabled analysis of performance and efficiency of machine-learning models on GPUs, contributing to understanding the solution's capabilities in real-time electronic component inspection.

5.1. Benchmarking and Testing

In order to quantitatively assess the performance of the presented solution in real-world production environments, we engineered an environment mirroring the configurations commonly found in our customers' setups. This environment was designed to incorporate 20 ASMT pick-and-place (P&P) machines, each mounting components and utilizing the gRPC protocol. The true litmus test for the system's performance can be seen in Figure 16, where the volume of access requests to TensorFlow for image processing describes the stress by the system under real-world conditions. This stress testing was important in gauging the solution's robustness and reliability in handling the demands of large-scale electronic component inspections. Over 1.6 million components were processed within a data frame over 12 hours.

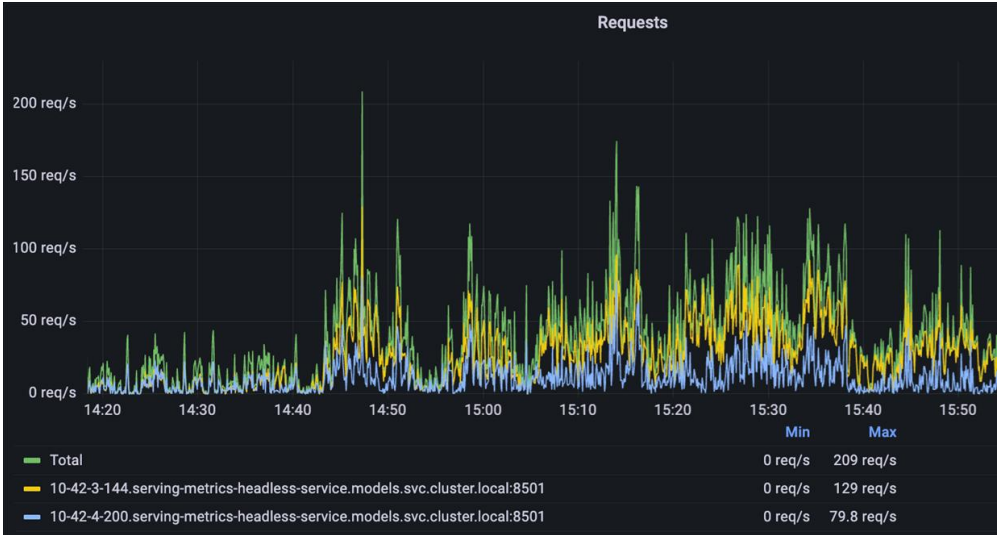


Figure 16. Illustrating the computational system's stress through the frequency of TensorFlow requests per second.

In addition to hardware evaluations, the software underwent thorough testing and refinement. Developed from the ground up using C++, our software underwent comparisons between different versions to ensure optimal performance and reliability. To further enhance the understanding of the presented solution's capabilities, we subjected our machine learning models to testing using TensorRT, an advanced deep learning inference optimizer and runtime library developed by NVIDIA. The evaluation of our models on GPUs, along with detailed performance metrics, is

presented in the following section, providing valuable insights into the efficiency and effectiveness of our solution in real-time electronic component inspection.

5.2. Pre-Processing Stage Evaluation

In the preprocessing stage, all images underwent initial processing for light balancing (clipping), aiming to address issues related to images with a high ratio of saturated pixels. These saturated pixels are indicators that the mounting machine can utilize to enhance the accuracy of component placement. See examples in Figure 12. In Figure 17, the frequency of clipped images during the system's operation on real-time data is depicted. It's evident that approximately 4.6 components per second are clipped in this scenario. However, as the system continuously conveys this information to the mounting machine, there is a reduction in this rate over time due to the feedback loop. This feedback mechanism contributes to improving the accuracy of placement and reducing the attrition rate in the assembly process.

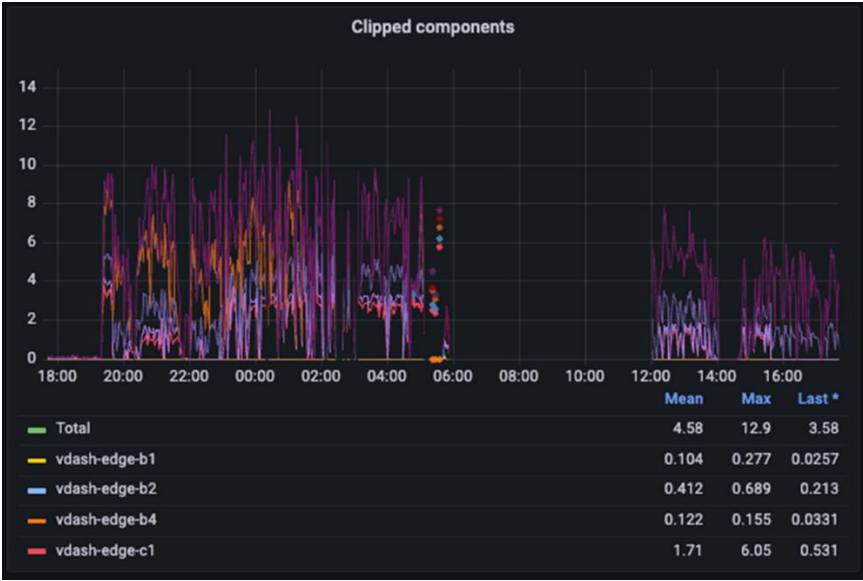


Figure 17. Clipped images processing with a high ratio of saturation.

In addition to addressing issues related to clipped images, the preprocessing stage also mitigates problems associated with blurry images, which can significantly impact mounting accuracy and increase attrition rates. Blurry images often occur due to incorrect programming that fails to allocate sufficient resolution (binning) to the image, resulting in a lack of sharpness (as observed in Figure 13). Another common cause of blurry images is incorrect programming of component height, leading to poor focus during image capture. Defective lighting or imaging systems can also contribute to image blurriness.

Figure 18 illustrates the occurrence of blurry images during system operation on real-time data. It's notable that the rate of blurry images is significantly lower compared to clipped images. This discrepancy underscores the importance of addressing issues related to image blurriness, albeit less frequent than clipped images, as they still pose a considerable risk to mounting accuracy and attrition rates. Addressing the root causes of blurry images through appropriate programming adjustments and system maintenance is crucial for improving overall assembly quality and reducing component attrition.



Figure 18. Blurry images processing due to inaccurate height programming or inaccurate focus issues.

5.3. Pre-Processing Stage – Component Dimensions Measurement

In high-volume production lines, accurately measuring the dimensions of mounted components is crucial due to the wide variety of component sizes encountered. Figure 19 displays the calculated width and height of mounted components over a 12-hour interval during testing. The data reveals that small passive components such as MLCCs and chip resistors are the most prevalent size category. We've observed a positive correlation between the size of the image and the end-to-end processing time, indicating that larger images take longer to process.

Obtaining real-life dimensions of components and calculating the mean and variance of their sizes are vital steps in improving the performance of pick-and-place (P&P) and Automated Optical Inspection (AOI) systems. These measurements provide accurate feedback for programming component dimensions, enhancing the software's performance by relying on real measurements of specific components rather than generic information from tables or specification sheets.

Furthermore, analyzing the variance in component sizes is essential for assessing the quality of materials used and the consistency of components within each container. Variance measurements help gauge how similar the components are within a batch, providing insights into manufacturing quality and potential issues related to component variation.

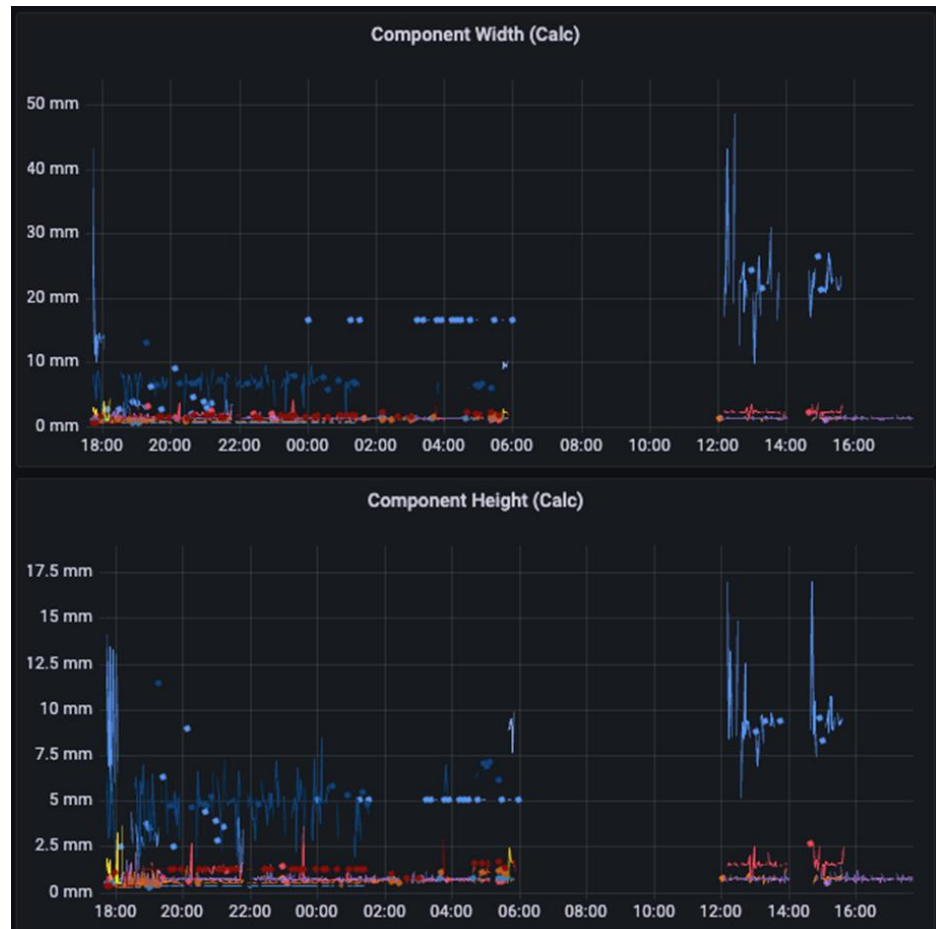


Figure 19. The calculated dimensions of the components exemplifying the various components processed in the production line.

5.4. Processing Stage – Component Defects and Corrosion Detection Algorithms

The experiment involved running both the defect and corrosion detection algorithms simultaneously while 20 machines operated asynchronously in a real-world production environment. The primary focus was to evaluate the processing time of these algorithms under high-volume conditions. Figure 20 illustrates the processing time of both algorithms over the duration of the test. During the experiment, the mean processing time for defect and corrosion detection algorithms was approximately 1.5 milliseconds per component and approximately 20 components per second. Note that the presented processing time solely accounts for the time spent on the detection algorithms and excludes preprocessing steps and network delays. These results demonstrate the efficiency and effectiveness of the implemented algorithms in real-time defect and corrosion detection. The size of the components significantly impacts processing performance. Figure 21 illustrates the diversity of component sizes processed during the test, highlighting the range of dimensions encountered during real-world production scenarios.



Figure 20. (top) Processing time of the defects detection algorithm, (bottom) processing time of the corrosion detection algorithm. Both running in parallel on the same images.

Table 1. Algorithm and system performance of old architecture with new.

Software Version	CPU	RAM (MB)	throughput (c/s)
Edge (C++)	0.1	25	460
Edge (Py)	0.14	123	4.6

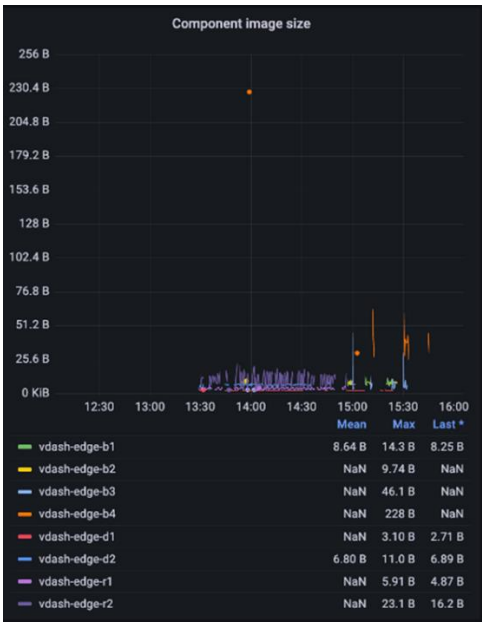


Figure 21. Variety of component sizes processed during the test, showcasing the range of dimensions encountered in real-world production scenarios.

5. Conclusion

We have introduced a groundbreaking method for real-time processing of images during assembly, marking a significant advancement in quality control within manufacturing environments. This novel approach revolutionizes the inspection process by enabling 100% inspection of all components, virtually eliminating visible defective components from the final products through an inline process.

By leveraging the time window between component pick-up and mounting, our method ensures realtime image processing, with an average turnaround time of less than 5 milliseconds. This efficiency is made possible by utilizing a fast network protocol, gRPC, and employing Kubernetes orchestration for seamless coordination across production environments. The implementation of C++ programming, real-time TensorFlow, and a cost-effective GPU further enhances the method's speed and reliability.

Through rigorous testing on 20 live production machines, we have validated the scalability and effectiveness of our approach in real-world manufacturing settings. The presented method represents a significant step forward in enhancing manufacturing quality control processes, offering unparalleled accuracy, speed, and efficiency in defect detection and elimination. Furthermore, the presented method ensures compliance with IPC-A-610, and IPC-STD-J-001 standards, highlighting its commitment to upholding industry-leading quality standards and regulations.

References

1. A. Goel and R. J. Graves, "Electronic system reliability: collating prediction models," *IEEE Transactions on Device and Materials Reliability*, vol. 6, no. 2, pp. 258–265, 2006.
2. B. Foucher, J. Boulli E E A, B. Meslet, and D. Das, "A review of reliability prediction methods for electronic devices." [Online]. Available: www.elsevier.com/locate/microrel
3. N. Nikolov, T. Papanchev, and A. Georgiev, "Reliability assessment of electronic units included in complex electronic systems," in *2017 40th International Spring Seminar on Electronics Technology (ISSE)*, IEEE, 2017, pp. 1–6.
4. M. Pecht, T. Shibutani, and L. Wu, "A reliability assessment guide for the transition planning to lead-free electronics for companies whose products are RoHS exempted or excluded," *Microelectronics Reliability*, vol. 62, pp. 113–123, Jul. 2016, doi: 10.1016/j.microrel.2016.03.020.
5. S. Cheng, C. M. Huang, and M. Pecht, "A review of lead-free solders for electronics applications," *Microelectronics Reliability*, vol. 75, Elsevier Ltd, pp. 77–95, Aug. 01, 2017, doi: 10.1016/j.microrel.2017.06.016.
6. C. M. Huang, J. A. Romero, M. Osterman, D. Das, and M. Pecht, "Life cycle trends of electronic materials, processes and components," *Microelectronics Reliability*, vol. 99, pp. 262–276, Aug. 2019, doi: 10.1016/j.microrel.2019.05.023.
7. D. Varna and V. Abromavičius, "A System for a Real-Time Electronic Component Detection and Classification on a Conveyor Belt," *Applied Sciences (Switzerland)*, vol. 12, no. 11, Jun. 2022, doi: 10.3390/app12115608.
8. I. Atik, "Classification of Electronic Components Based on Convolutional Neural Network Architecture," *Energies (Basel)*, vol. 15, no. 7, Apr. 2022, doi: 10.3390/en15072347.
9. W. Zhao, S. R. Gurudu, S. Taheri, S. Ghosh, M. A. M. Sathiaselalan, and N. Asadizanjani, "PCB Component Detection Using Computer Vision for Hardware Assurance," *Big Data and Cognitive Computing*, vol. 6, no. 2, Jun. 2022, doi: 10.3390/bdcc6020039.
10. M. A. Mallaiyan Sathiaselalan, O. P. Paradis, S. Taheri, and N. Asadizanjani, "Why is deep learning challenging for printed circuit board (Pcb) component recognition and how can we address it?," *Cryptography*, vol. 5, no. 1, 2021, doi: 10.3390/cryptography5010009.

11. M. M. Tehranipoor, U. Guin, and D. Forte, "Counterfeit integrated circuits," in *Counterfeit Integrated Circuits*, Springer, 2015, pp. 15–36.
12. S. P. Gayialis, E. P. Kechagias, G. A. Papadopoulos, and D. Masouras, "A Review and Classification Framework of Traceability Approaches for Identifying Product Supply Chain Counterfeiting," *Sustainability (Switzerland)*, vol. 14, no. 11. MDPI, Jun. 01, 2022. doi: 10.3390/su14116666.
13. a S. Report, "counterfeit parts : increasing Awareness and Developing countermeasures Counterfeit Parts : Increasing Awareness and," *T*, no. March, 2011.
14. L. W. Kessler and T. Sharpe, "FAKED PARTS DETECTION The counterfeiting of electronics components and related parts is widespread. But newly developed methods promise now to help identify counterfeit plastic-encapsulated components using detection methods that cannot be tricked. PLUS: ," *Printed Circuit Design & Fab*, vol. 27, no. 6, p. 64, 2010.
15. U. Guin and M. Tehranipoor, "On selection of counterfeit IC detection methods," in *IEEE north atlantic test workshop (NATW)*, 2013.
16. E. Weiss, "Counterfeit mitigation by in-line deep visula inspection," SMTA. [Online]. Available: <http://iconnect007.uberflip.com/i/1440051-smt007-jan2022/87?>
17. and G. G. E. Weiss Dr, E. Zeev, "System and method for detection of counterfeit and cyber electronic components," WO2020202154A1, 2019
18. A. Spieske and H. Birkel, "Improving supply chain resilience through industry 4.0: a systematic literature review under the impressions of the COVID-19 pandemic," *Comput Ind Eng*, vol. 158, p. 107452, 2021.
19. G. M. Razak, L. C. Hendry, and M. Stevenson, "Supply chain traceability: a review of the benefits and its relationship with supply chain resilience," *Production Planning and Control*, vol. 34, no. 11, pp. 1114–1134, 2023, doi: 10.1080/09537287.2021.1983661.
20. E. Weiss, "Electronic component solderability assessment algorithm by deep external visual inspection," in *2020 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, IEEE, 2020, pp. 1–6.
21. E. Weiss and Z. Efrat, "System and method for nondestructive assessing of solderability of electronic components," P-603537-PC, 2021
22. E. Weiss, "Revealing Hidden Defects in Electronic Components with an AI-Based Inspection Method: A Corrosion Case Study," *IEEE Trans Compon Packaging Manuf Technol*, 2023, doi: 10.1109/TCPMT.2023.3293005.
23. E. Weiss, "Preventing Corrosion-related Failures in Electronic Assembly: A Multi-case Study Analysis," *IEEE Trans Compon Packaging Manuf Technol*, May 2023, doi: 10.1109/TCPMT.2023.3285776.
24. E. Weiss, "AI Detection of Body Defects and Corrosion on Leads in Electronic Components, and a study of their Occurrence," in *2022 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, IEEE, 2022, pp. 1–6.
25. E. Weiss, "Detecting Corrosion to Prevent Cracks in MLCCs with AI," *Journal of Failure Analysis and Prevention*, 2023, doi: 10.1007/s11668-023-01824-3.
26. T. I. Băjenescu, T.-M. I. Băjenescu, and M. I. Băzu, *Component reliability for electronic systems*. Artech House, 2010.
27. K. Choudhary and P. Sidharthan, "Reliability prediction of Electronic Power Conditioner (EPC) using MIL-HDBK-217 based parts count method," in *2015 International Conference on Computer, Communication and Control (IC4)*, IEEE, 2015, pp. 1–4.
28. V. Forti, K. Baldé, and R. Kuehr, "E-waste statistics: guidelines on classifications, reporting and indicators," 2018.

29. N. Singh and O. A. Ogunseitan, "Disentangling the worldwide web of e-waste and climate change co-benefits," *Circular Economy*, vol. 1, no. 2, p. 100011, Dec. 2022, doi: 10.1016/j.cec.2022.100011.
30. V. Forti, C. P. Balde, R. Kuehr, and G. Bel, "The Global E-waste Monitor 2020: Quantities, flows and the circular economy potential," 2020.
31. A. Esfandyari, S. Härter, T. Javied, and J. Franke, "A lean based overview on sustainability of printed circuit board production assembly," *Procedia CIRP*, vol. 26, pp. 305–310, 2015.
32. V. Tripathi, S. Chattopadhyaya, A. K. Mukhopadhyay, S. Sharma, C. Li, and G. Di Bona, "A Sustainable Methodology Using Lean and Smart Manufacturing for the Cleaner Production of Shop Floor Management in Industry 4.0," *Mathematics*, vol. 10, no. 3, Feb. 2022, doi: 10.3390/math10030347.
33. "IPC-A-610 Acceptability of Electronic Assemblies," 2017. [Online]. Available: www.ipc.org
34. E. Weiss, "System and method for detection of counterfeit and cyber electronic components," 2019
35. Z. Peric, M. Savic, M. Dincic, N. Vucic, D. Djosic, and S. Milosavljevic, "Floating point and fixed point 32-bits quantizers for quantization of weights of neural networks," in *2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, IEEE, 2021, pp. 1–4.
36. N. Wang, J. Choi, D. Brand, C.-Y. Chen, and K. Gopalakrishnan, "Training deep neural networks with 8-bit floating point numbers," *Adv Neural Inf Process Syst*, vol. 31, 2018.
37. J. Liu *et al.*, "Towards provably efficient quantum algorithms for large-scale machine-learning models," *Nat Commun*, vol. 15, no. 1, p. 434, 2024.
38. Y. Liao, Y. Xu, H. Xu, Z. Yao, L. Wang, and C. Qiao, "Accelerating federated learning with data and model parallelism in edge computing," *IEEE/ACM Transactions on Networking*, 2023.

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