

Article

Not peer-reviewed version

Computer Vision-Based Product Quality Inspection and Novel Counting System

[CHANG-HYUN LEE](#), [YUNSIK KIM](#), [HUNKEE KIM](#) *

Posted Date: 4 November 2024

doi: 10.20944/preprints202411.0133.v1

Keywords: SIFT Algorithm; image-based quality inspection; product counting; YOLO v8 Pose; load cell; feature matching



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.

Article

Computer Vision-Based Product Quality Inspection and Novel Counting System

CHANGHYUN LEE, YUNSIK KIM and HUNKEE KIM *

Inha University, AI System Lab, Department of Advanced Materials Processing Engineering, 36 Gaetbeol-ro, Yeonsu-gu, Incheon, Republic of Korea; ckd9848@gmail.com (C.L.); yskim9701@naver.com (Y.K.)

* Correspondence: hunkeekim@inha.ac.kr

Abstract: In this study, we aim to enhance the accuracy of product quality inspection and counting in the manufacturing process by integrating image processing and human body detection algorithms. We employed the SIFT algorithm combined with traditional image comparison metrics such as SSIM, PSNR, and MSE to develop a defect detection system that is robust against variations in rotation and scale. Additionally, the YOLO v8 Pose algorithm was used to detect and correct errors in product counting caused by human interference on the load cell in real-time. By applying the image differencing technique, we accurately calculated the unit weight of products and determined their total count. In our experiments conducted on products weighing over 1kg, we achieved a high accuracy of 99.268%. The integration of our algorithms with the load cell-based counting system demonstrates reliable real-time quality inspection and automated counting in manufacturing environments.

Keywords: SIFT Algorithm; image-based quality inspection; product counting; YOLO v8 Pose; load cell; feature matching

1. Introduction

With the recent advancement of automation and AI (artificial intelligence) across industries, these technologies have become essential for enhancing productivity and reducing manufacturing costs. In the manufacturing process, tasks such as defect detection, quality control, and inventory management must be carried out in real time through automated systems, where high accuracy and efficiency are essential [1]. Especially in manufacturing sectors that adopt a multi-product flexible production system, the ability to flexibly produce a variety of products is required, demanding precise adjustments whenever environmental conditions change. Given the nature of the manufacturing industry, new data collection and model training are required whenever a new product is introduced or environmental conditions shift, leading to significant costs in technology and human resources. Computer vision addresses these challenges by detecting defects through image analysis and performing precise quality inspections, making it well-suited for environments that require flexible, multi-product production. In particular, computer vision can adapt to environmental changes without requiring new data collection or model retraining, allowing manufacturers operating multi-product flexible production systems to quickly adapt to new product launches or changes in environmental conditions without additional learning processes. Through these capabilities, computer vision technology maximizes both precision and productivity by detecting defects through product image analysis and enabling rapid and precise quality inspection, while also contributing to savings in technological and human resources [2–4]. The application of this technology is expected to play a pivotal role in identifying various defects through product image analysis and conducting fast and precise quality inspections. Key techniques include the SSIM (Structural Similarity Index Measure), MSE (Mean Squared Error), and PSNR (Peak Signal-to-Noise Ratio), which are mainly used to evaluate structural similarity and pixel differences between images [5–7]. Notably, SSIM has been proven effective in resolving ambiguity when tracking multiple

objects, making it useful not only for quality assessment but also for tasks requiring precise object recognition [8].

However, there are several challenges that must be addressed in order to build such a system. First, methods like SSIM, PSNR, and MSE are not sensitive to image rotation or positional changes. This means that if the angle or size of an image varies, defects may not be accurately detected. Second, load cell-based weight measurement systems frequently encounter errors that prevent precise product counting due to resolution limitations. For instance, if a load cell measures weight in 500g increments, over-counting or under-counting may occur if the actual weight doesn't align precisely with these units. Third, errors can arise when workers inadvertently affect the load cell while loading products. In manufacturing environments, if a worker steps onto or approaches the load cell while placing a product, the load cell may register the person's weight as well, leading to inaccurate counts. For example, if a worker briefly steps on the load cell while placing a product, the system may register both the worker's and the product's weights, resulting in incorrect calculations. This lowers the reliability of the load cell measurement system and impacts inventory management. These issues can seriously disrupt real-time counting systems, and if left unaddressed, the accuracy of inventory management and quality control cannot be assured. Specifically, when the weights of people and products cannot be differentiated, inventory calculations may be inaccurate, reducing the overall efficiency of the manufacturing process.

This study was developed to improve the automation and accuracy of defect detection, quality control, and inventory management by integrating SIFT(Scale-Invariant Feature Transform)-based defect detection, a real-time counting correction system using YOLO v8 pose, and a precision counting mechanism using difference image techniques. The system is designed to solve the consistency, time and labor cost issues caused by manual inspection in large-scale manufacturing environments. Product defects are inspected through computer vision technology using cameras, and if the product has a defect and the detected value is different from the standard product, the system notifies the operator for reprocessing. Reprocessed products are re-examined using differential imaging technology to ensure they meet quality standards. Test results showed that the system demonstrated high accuracy in defect detection and quality control, contributing to reducing human errors and significantly improving the efficiency of the overall manufacturing process.

Figure 1 presents a schematic diagram of the entire system. When a product is placed on the conveyor belt, defects are detected, and uniformity is assessed through a camera installed on the belt, after which accepted products proceed to the counter. At this stage, Camera 2 identifies the products, calculates the quantity through difference images, and computes the average value. Weight measurement and counting are then performed to ensure accurate inventory management. This process addresses the issues mentioned above, and in this study, we propose three improved technologies. First, we utilize the SIFT algorithm to compare images and detect defects that remain robust against rotation and size changes in product images. SIFT extracts feature points from an image, enabling accurate similarity analysis regardless of rotation or size [9–13]. Recent studies demonstrate SIFT's high efficacy in modern applications, thanks to continuous improvements in memory storage. Advances in data compression, for instance, allow consecutive nibble pairs to be stored within a single byte, reducing memory usage by half without causing alignment issues. This bit-level improvement supports faster comparative analysis while preserving storage efficiency and matching accuracy [14]. After correcting the product's size and orientation using SIFT, product defects and uniformity are determined by analyzing the SSIM, PSNR, and MSE values with the $S_{combined}$ formula and difference image technique. Second, we introduce the YOLO v8 pose algorithm to create a system that corrects counting errors in real-time whenever an operator is detected on the load cell. This solution ensures accurate product counting by temporarily pausing weight measurement when an operator steps onto or approaches the load cell. Third, we develop an accurate product counting method using the difference image technique to address errors caused by the load cell's resolution limitations. During initial setup, a specific number of products are placed on the load cell, detected through difference images, and then the unit weight is calculated. Product

count is subsequently derived based on the total weight. This approach minimizes counting errors arising from product placement or external factors.

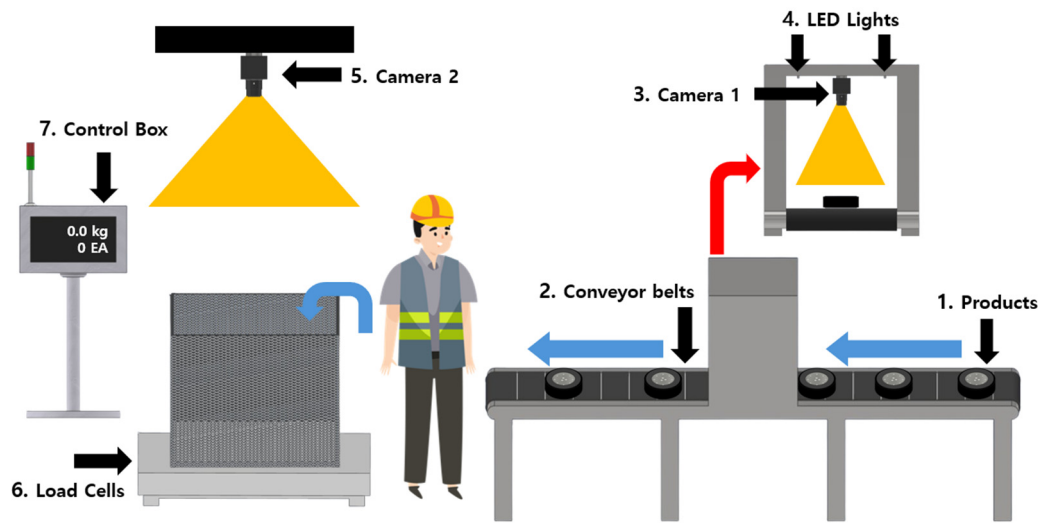


Figure 1. This figure is a schematic image of the entire system.

The remainder of this paper is structured as follows: Section 1 provides an introduction, detailing the study’s purpose and scope. Section 2 reviews related research, focusing on advancements in computer vision-based defect detection, body motion detection, and automated inventory management. Section 3 outlines the methods proposed in this paper for product defect detection, uniformity assessment, worker recognition, precision counting, and inventory management. Section 4 presents the experimental setup and results, analyzing the performance of the proposed techniques. Finally, Section 5 concludes the paper with a summary of the findings and recommendations for future research directions.

2. Related Works

2.1. Computer Vision-Based Defect Detection

2.1.1. Limitations of Deep Learning Techniques Based on Image Classification

Recently, deep learning technology has made great progress and is being used in various fields. In particular, deep learning models such as CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network) are also attracting attention in the field of image classification. Additionally, the recent emergence of new architectures such as vision transformer (ViT) expands image processing possibilities, providing CNN-like performance. However, while this approach shows excellent performance, it still has several limitations.

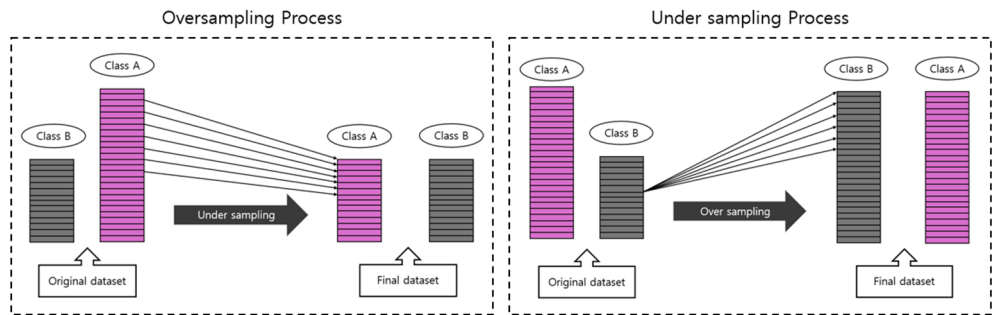


Figure 2. This figure shows how to amplify and reduce data to address imbalances in a dataset, with over sampling on the left and under sampling on the right.

First, there is a class imbalance problem. In most datasets, the imbalance between majority and minority classes has a negative impact on the performance of deep learning models [15]. In particular, when class imbalance is severe, feature learning of minority classes is not performed properly, and the model tends to perform excellently only in predictions for majority classes [16]. In a recent study, the performance degradation of deep learning models for various class imbalance problems was analyzed, and it was found that the larger the imbalance, the greater the tendency for model accuracy to deteriorate [17,18]. To solve this problem, use random sampling technology to under sample the majority category data to fit the minority category data, or use a technology called GAN (Generative Adversarial Nets) to oversample the minority category data. Even if these techniques are used, accurate measurement is difficult [19]. Second, deep learning models such as CNN have high computational complexity and require a lot of resources during the learning and inference process [18]. Especially in a real-time environment, this requires research to optimize the computational load for real-time product quality evaluation using multiple frame rate cameras and various deep learning models [20,21]. CNN can extract various spatial features of an image through multiple layers of convolution, but its computational cost is very high due to its structural characteristics. This can cause difficulties in applying deep learning models when real-time processing is required in an actual industrial environment, and in order to solve these problems, several studies have argued that lightweight network design or hardware acceleration techniques are needed. . Third, using deep learning models in a manufacturing environment is subject to various limitations. Due to the nature of the manufacturing industry, the model must be retrained every time a new product is manufactured in addition to an existing product, which inevitably requires significant time and cost during the data collection and labeling process. Additionally, if the manufacturing facility or environment changes, the measurement environment must be rebuilt and the model must be retrained, which can have a negative impact on operational efficiency.

To solve these limitations, this paper uses computer vision techniques such as SSIM, MSE, PSNR, and SIFT. and compare the image data. The goal is to compare good and defective products without the need for a predefined learning data set.

2.1.2. Application of Vision Algorithm in Product Classification

The development of vision algorithms is currently making an important contribution to the detection and classification of product defects and quality control in the manufacturing industry, and continues to develop in the direction of increasing real-time processing capabilities and accuracy. Through this, it plays an important role in maximizing productivity and quality control efficiency in the manufacturing process. When applying vision algorithms in product classification, commonly used image comparison techniques such as SSIM, MSE, and PSNR are used. This is a technology that evaluates structural similarity and pixel differences between the same images. Although these techniques are useful for basic image quality evaluation, they have the limitation of not being sufficiently robust against rotation or size changes.

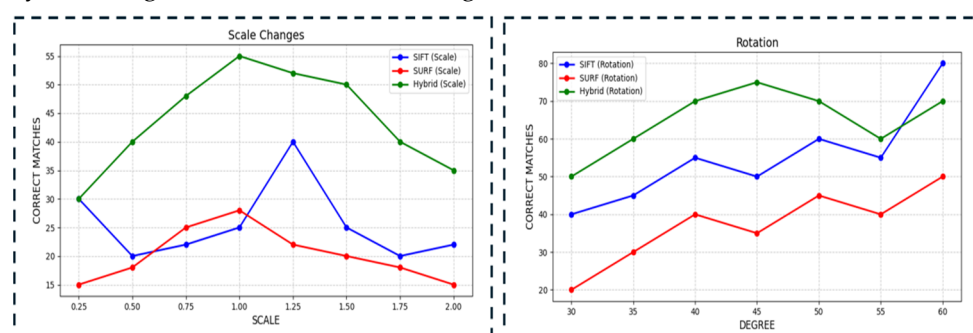


Figure 3. This figure shows a comparative analysis of the performance of the SIFT, SURF, and hybrid algorithms. On the left is the performance metric for scale change, and on the right is the performance metric for rotation.

In contrast, feature-based algorithms such as SIFT can extract feature points robustly even when image rotation or size changes, enabling accurate defect detection [22,23]. SIFT has the advantage of being able to extract feature points stably despite rotation or size changes within the image [24,25]. According to research results, when testing SURF (Speed-Up Robust Features), SIFT, and Hybrid technologies, hybrid was superior in scale changes, but SIFT was found to provide more accurate detection when detecting features such as rotation invariance [26]. Based on this, this paper aims to develop a model that increases the recognition rate of products and enables more accurate defect detection by utilizing the SIFT technique to overcome the limitations of existing techniques.

2.2. Research on Body Movement Detection Based on Body Skeletal Structure

2.2.1. Latest Research Trends in Body Movement Detection

Body movement detection is advancing as a technology that leverages deep learning, a field within machine learning, to identify human movement, detect hazards, and recognize specific actions. In HAR (Human Activity Recognition), deep learning models play a crucial role in accurately recognizing various movements by analyzing video and sensor data. The development of HAR technology primarily involves the integration of diverse deep learning architectures [27]. Applications such as real-time hand gesture recognition are made possible by these advancements, and research has even extended to gesture recognition based on body skeletal structure [28]. These studies highlight the impressive capabilities of deep learning in tracking and recognizing body movements in real time and suggest which approaches may be more effective in particular situations by comparing different architectures.

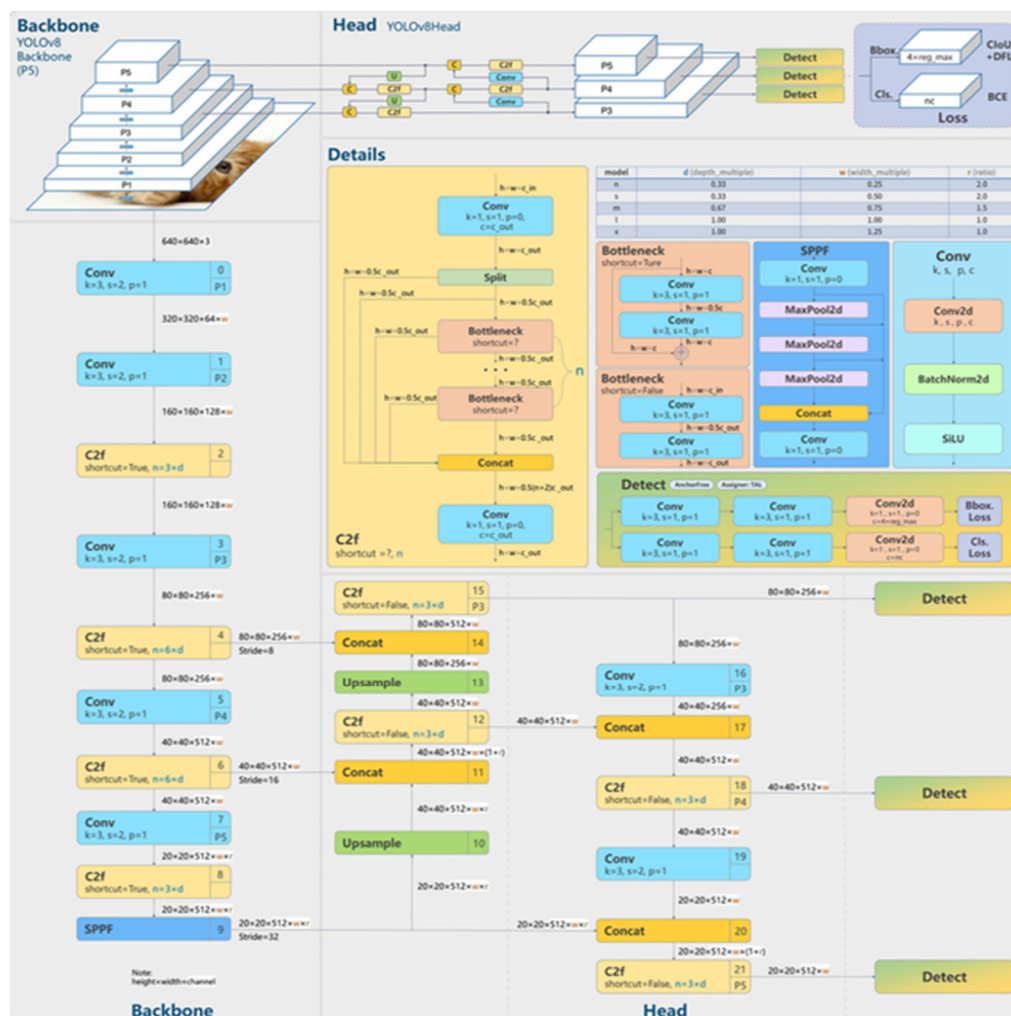


Figure 4. This figure is an image of the YOLO V8 architecture to illustrate the overall system. [29].

One such architecture, the YOLO v8 pose model, incorporates advanced features for pose estimation, offering enhanced speed and accuracy over previous versions. It utilizes the new C2f module, which allows for lightweight computation and efficient feature reuse, optimizing the model for dense and complex tasks like pose estimation. Unlike previous YOLO models, YOLO v8 is anchor-free, simplifying the detection process. This architecture includes a streamlined backbone—the core neural network responsible for feature extraction—enabling more efficient keypoint detection. These improvements make YOLO v8 particularly well-suited for real-time applications such as human motion tracking [30].

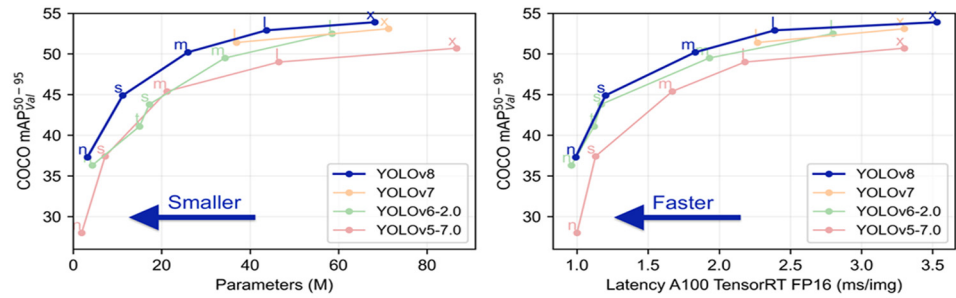


Figure 5. This figure shows a graph comparing the performance of the YOLO object detection model. [31].

The following summarizes the performance evolution across YOLO versions from v1 to v8. YOLOv1 introduced real-time object detection with a single network pass but struggled with small object detection. YOLOv2 improved accuracy with anchor boxes and batch normalization, while YOLOv3 enhanced multi-scale object detection by adding the Darknet-53 backbone and multi-scale prediction capabilities. YOLOv4 achieved a balance between speed and accuracy through CSPDarknet-53, SPP, and PAN modules, and YOLOv5 increased user-friendliness and learning efficiency by transitioning to PyTorch. Building on this, YOLOv6 introduced RepVGG blocks and a decoupled head structure for better performance, and YOLOv7 achieved the highest accuracy and speed by maximizing computational efficiency with E-ELAN and RepConv [32]. The latest version, YOLO v8, represents the most advanced iteration, enhancing speed and accuracy with an anchor-free design and the C2f module. For these reasons, this study selects YOLO v8 as the basis for further research and experimentation.

2.3. Automated Inventory Management and Product Counting System

2.3.1. Inventory Management and Product Counting Market Trends

Recently, research and advancements in inventory management and product counting systems have gained attention as key components of smart factory implementation [33]. In manufacturing, intelligent systems are essential to maximize inventory management efficiency and enhance the accuracy of product counting [34,35]. According to a survey, technologies such as barcodes, QR codes, AI, cloud computing, IoT (Internet of Things), RFID (radio frequency identification), and WMS (warehouse management systems) are widely used.

Barcode and QR code technologies provide cost-effective and reliable methods for tracking products throughout the supply chain. Barcodes are scanned at various production and distribution stages for quick identification and status updates, while QR codes, with their higher data capacity, allow for the inclusion of product specifications or batch details and are easily integrated into modern systems. In addition, AI, cloud computing, IoT, and RFID technologies have become central to improving inventory management and product counting in contemporary manufacturing systems [36]. AI-based systems, in particular, are effective in reducing human error and enhancing accuracy by using computer vision and data analytics to streamline product calculations and inventory management, supporting real-time tracking and automated decision-making.

WMS further enhance these processes by providing centralized control and automation for tracking and managing inventory. These technologies demonstrate a significant impact on the

accuracy and efficiency of inventory management and product counting. Moreover, automated warehouse operation systems improve inventory management precision even across multiple products and varied conditions, ultimately boosting productivity and reducing costs.

2.3.2. Existing Product Counting Methods and Problems

The problem with existing load cells is that accurate product counting is impossible due to resolution limitations during the calculation process. For example, if the resolution of the load cell is rounded to 500g, over-counting or under-counting may occur if the actual weight deviates from the standard value. Additionally, in a manufacturing environment, if a worker temporarily stands on or near a load cell while loading product, the system may mistakenly recognize the worker's weight, leading to incorrect calculations. These inaccuracies have serious implications for inventory management and reduce the overall efficiency of the manufacturing process. To address these issues, research on smart load cells has shown that smart load cells achieve measurement errors of less than 100 g in industrial applications weighing up to 400 kg, providing more accurate readings than traditional systems [37]. However, despite these improvements, these systems are still limited in terms of maximum weight capacity and resolution, making them inadequate for use in a wide range of industrial applications.

In this paper, in order to improve the accuracy in the product counting and inventory management process, a system that recognizes when a worker approaches and sets the unit weight through the product counting value using a technology that does not affect the counting and a difference image technique Aim for development. Through this, we believe that it will contribute to reducing product counting errors and increasing the reliability of inventory management by securing the limitations of existing resolution.

3. Methods and Result

This study aims to evaluate the likelihood of a product being a normal or defective item by analyzing similarities between images. Precise image comparison analysis is essential for automated quality inspection in manufacturing processes, especially in detecting subtle differences. Higher SSIM and PSNR values indicate greater similarity between images, while a lower MSE value suggests a smaller difference, thus representing greater similarity. However, high SSIM and PSNR values and a low MSE value do not guarantee that two images will appear visually identical. These metrics reflect only specific aspects of image similarity and may not be sensitive to subtle differences or variations in object position and scale. For instance, identical objects may yield high SSIM and PSNR values even if the image is rotated or shifted, but they may still look visually different. Since SSIM reflects structural similarity and PSNR and MSE focus on pixel differences, their reliability may decrease when structural and detailed differences are mixed. Particularly, if a product is rotated or shifted, high SSIM, PSNR, and low MSE values may still exhibit significant visual discrepancies. To address this issue, this study applies the SIFT algorithm to detect the product's orientation first. SIFT identifies keypoints within an image and extracts features that are robust to rotation and translation, enabling the product to be restored to its original orientation even if positioned at various angles. This alignment allows more reliable use of SSIM, PSNR, and MSE.

Additionally, the $S_{combined}$ formula combines the values of SSIM, PSNR, and MSE to produce a final evaluation score, determining whether a product should be inspected. The $S_{combined}$ formula appropriately adjusts the weights of each metric, integrating diverse quality information that cannot be assessed with a single metric alone. This approach allows for clearer identification of potential defects when SSIM, PSNR, or MSE values exceed a threshold. Furthermore, if the $S_{combined}$ result deviates from a specific standard, the difference image technique is applied to visually confirm changes in the product. The difference image technique calculates pixel-level differences between two images, effectively highlighting surface defects or color changes. This method allows for accurate identification of defect locations, providing a basis for operators to address issues immediately if necessary. In conclusion, by using the SIFT algorithm to align the product's position and orientation,

applying the $S_{combined}$ formula to assess quality, and employing the difference image technique to visually detect minor defects, this study contributes to enhancing overall quality management.

3.1. Image-Based Product Quality Inspection and Feature Matching Techniques

3.1.1. Rotation and Scale Invariance in Image Comparison Using SIFT Algorithm

As shown in Figure 6, after loading two images, the SIFT (Scale-Invariant Feature Transform) algorithm is used to detect keypoints within the images and calculate descriptors for those keypoints. SIFT is an algorithm designed to detect keypoints that are invariant to scale, rotation, and illumination changes, enabling it to reliably find consistent features even under various transformations [38]. The first step of the SIFT algorithm is to locate keypoints in the scale space. This is achieved by using a Gaussian filter to process the image at different scales, allowing the detection of important keypoints by identifying extrema (maxima and minima) at each scale. The process of finding keypoints in the scale space, where a Gaussian blur is applied, utilizes the Difference of Gaussian (DoG) method [41]:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (1)$$

Here, $G(x, y, \sigma)$ represents the image with Gaussian blur applied at scale σ , $I(x, y)$ is the original image, and k denotes the scale factor. In this process, the differences between images at each scale are computed to identify extrema (maxima/minima), which are detected as keypoints [39–41]. By calculating the gradient magnitude and orientation of the pixels surrounding each keypoint, a principal orientation is assigned to each keypoint to ensure rotational invariance. The gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ are computed using the following equations:

$$m(x, y) = \sqrt{L(x+1, y)^2 - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2)$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right) \quad (3)$$

Here, $L(x, y)$ represents the intensity of the image, $m(x, y)$ is the gradient magnitude at that position, and $\theta(x, y)$ is the gradient orientation. Based on the calculated orientation information, a principal direction is assigned to each keypoint, enabling the keypoints to maintain rotational invariance [39–41]. To proceed with the matching process between the two images, the Euclidean distance between each descriptor is calculated to assess their similarity. The distance between two descriptors, p and q , is defined as follows:

$$d(p, q) = \sqrt{\sum_{i=1}^{128} (p_i - q_i)^2} \quad (4)$$

Here, p_i and q_i represent the i -th components of the two descriptors, respectively. The smaller the Euclidean distance, the more similar the two descriptors are considered, enabling the matching of keypoints between the two images. Once the matching is completed, the rotation angle between the two images can be estimated, and this angle is calculated using the following equation:

$$\theta = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (5)$$

Here, (x_1, y_1) and (x_2, y_2) represent the coordinates of the matched keypoint pairs in the two images, respectively. Using this equation, the rotation angle between the two images can be estimated, allowing for rotation correction and image restoration based on this information. Additionally, by calculating the distance between the matched keypoints, the scale variation between the two images can be assessed, enabling the identification of whether the image has been scaled up or down.

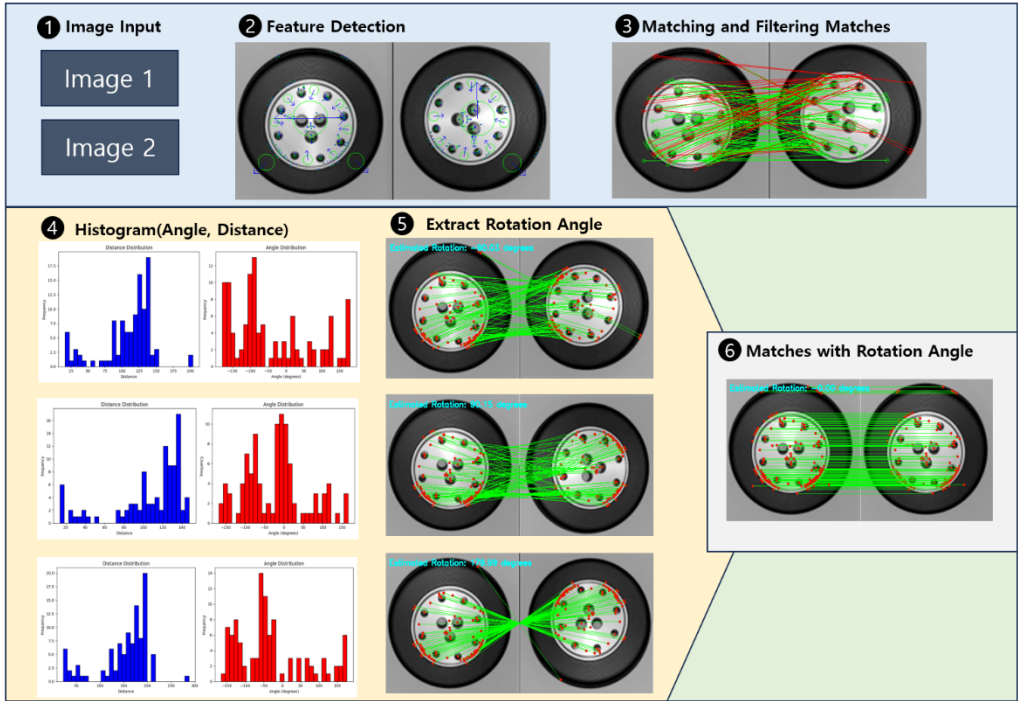


Figure 6. This figure shows an example of extracting descriptors from two input images using the SIFT algorithm, where one image serves as the reference. The algorithm restores the original rotation of the second image based on the reference, illustrating the process of image alignment.

3.1.2. Defect Detection Using a Combined SSIM, PSNR, and MSE Evaluation

Figure 7 shows the difference images between A product and B product obtained through pre-processing with SIFT, as well as the difference image between the two. To quantitatively evaluate the similarity between the two images, SSIM (Structural Similarity Index), PSNR (Peak Signal-to-Noise Ratio), and MSE (Mean Squared Error) were applied. These metrics are used to accurately compare and detect defects between normal and defective products, enabling efficient identification of product defects. The restored images were evaluated using SSIM, PSNR, and MSE to quantitatively assess the similarity between the two images, with the formulas for each method shown below [42–44]:

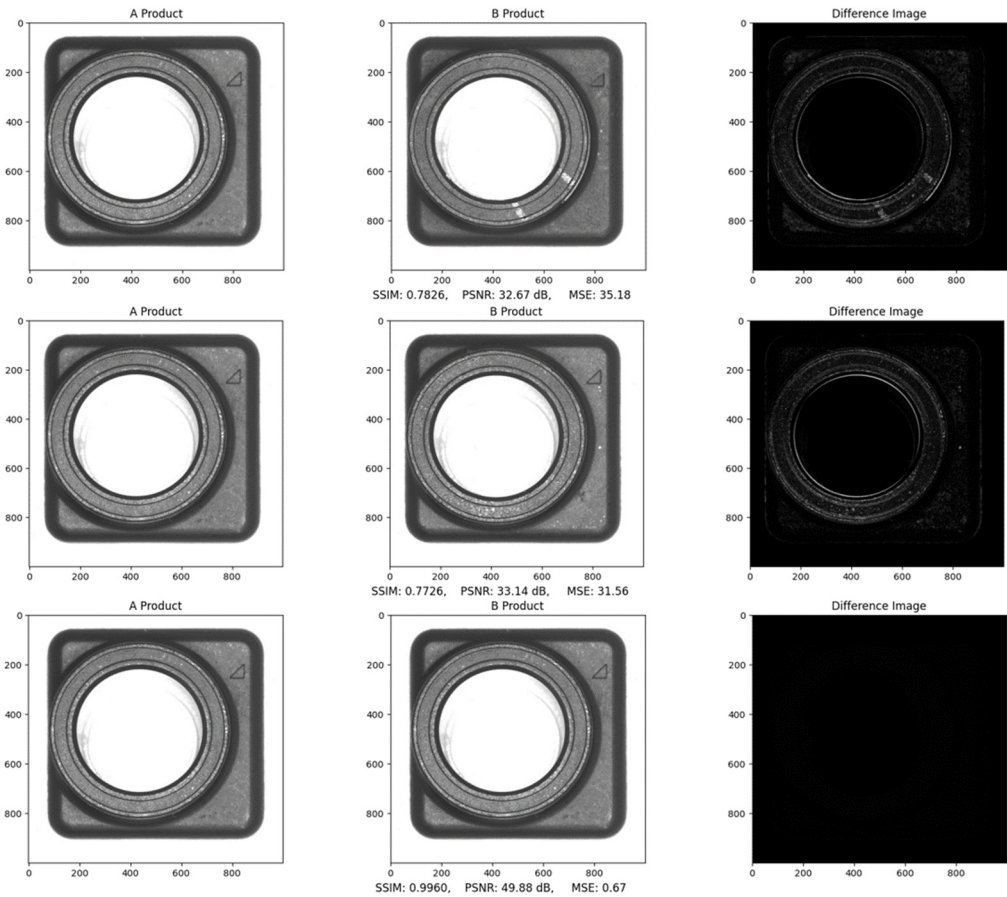


Figure 7. This figure shows the difference images for products A and B, along with the numerical results of SSIM, PSNR, and MSE derived from the difference image analysis.

Table 1. Formulas and Descriptions for SSIM, MSE, and PSNR Metrics.

Metric	Formula
SSIM	$\frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (6)$ μ_x and μ_y: The mean brightness values of the two images σ_x^2 and σ_y^2: The variance of each image σ_{xy}: The covariance between the two images C_1 and C_2: Constants for stability
MSE	$\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n [I_1(i,j) - I_2(i,j)]^2 \quad (7)$ $I_1(i,j)$ and $I_2(i,j)$: The pixel values at the i,j coordinates of the two images m and n: The dimensions of the image
PSNR	$10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (8)$ MAX_I: The maximum possible pixel value in the image MSE: The Mean Squared Error between the two images

By applying SSIM, PSNR, and MSE to the restored images, a quantitative evaluation was performed based on structural similarity, signal-to-noise ratio, and mean squared error between the two images. SSIM measures the structural similarity of the images, PSNR assesses the signal-to-noise ratio, and MSE analyzes the detailed pixel differences, enabling an overall evaluation of similarity. These metrics can vary according to specific criteria set by the user and may be interpreted differently

depending on the goals, application area, and quality requirements of the image processing task. Therefore, users can establish standards for each metric according to the project's objectives and requirements and evaluate image similarity based on these standards. For example, in manufacturing processes where defect detection is critical, even minor defects may significantly impact results. Thus, an SSIM value of 0.95 or higher could indicate an acceptable product, and a PSNR value of 40 dB or higher could suggest good quality [42]. In the case of MSE, a lower value signifies fewer differences between the two images. Therefore, to ensure consistent interpretation within $S_{combined}$, the inverse of MSE is used in calculations. This approach allows for a more intuitive interpretation, where a higher $S_{combined}$ score indicates greater similarity between the images. Finally, the similarity score $S_{combined}$, reflecting the weights of SSIM, PSNR, and MSE, is calculated as follows:

$$S_{combined} = \omega_{SSIM} * S_{SSIM} + \omega_{PSNR} * S_{PSNR} + \omega_{MSE} * \frac{1}{S_{MSE}} \quad (9)$$

The weights ω_{SSIM} , ω_{PSNR} , ω_{MSE} in the $S_{combined}$ formula sum to 1 and are adjusted based on the characteristics that each metric evaluates. This adjustment is not just about evaluating the images, but also about focusing on the specific defects and characteristics of the product. First, if the overall appearance or structure of the product is important, the weight of the SSIM metric, ω_{SSIM} , is increased. SSIM evaluates the structural similarity of the image, focusing on structural elements such as patterns, edges, and textures of the product. This is suitable in situations where structural defects are critical, such as in the bending of metal products or the consistency of patterns in textiles. Second, when noise or distortion on the product's surface is the focus, the weight of the PSNR metric, ω_{PSNR} , is increased. PSNR plays a significant role in examining surface scratches or the finishing of lens surfaces. Third, when fine pixel-level defects are of particular importance, the weight of the MSE metric, ω_{MSE} , is increased. MSE precisely calculates the differences between pixels, making it ideal for processes that need to detect very small defects. In conclusion, the weights in the $S_{combined}$ formula are set according to the product's characteristics and the type of defects being emphasized. By adjusting the weights according to the specific features of each metric, the efficiency and accuracy of defect detection can be improved.

3.2. Product Counting Algorithm Using Camera-Based Skeleton Tracking and Body Part Detection

3.2.1. Classification of Counting Classes Based on Body Part Detection

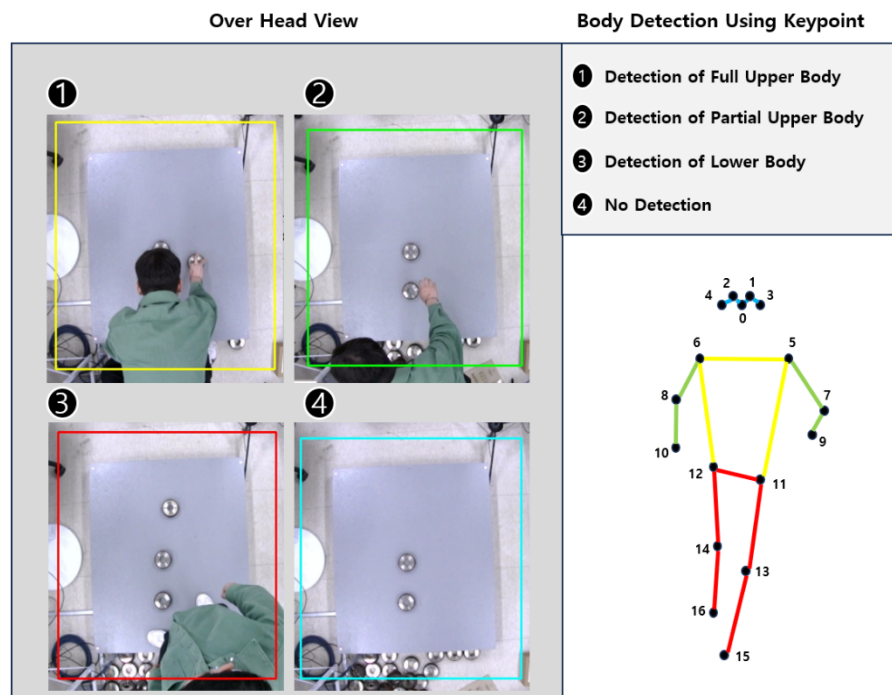


Figure 8. This figure shows the case-based operation in the load cell counting system using a body detection algorithm.

In this study, we developed a product counting system that applies the YOLO v8 Pose algorithm to detect the human body and correct errors that occur when a person steps onto the load cell [45,46]. The core of the research lies in using a camera and algorithm to detect human body parts in real time, distinguishing factors that affect weight data measured by the load cell, and reducing counting errors. Body parts that influence load cell weight measurements were categorized into four classes, and the load cell’s weight measurement actions were controlled according to each class.

- **Full Upper Body Detection:** When the full upper body is detected within the load cell area, weight measurement is temporarily paused. This is because the structure, in which the camera views the load cell from above, may result in the lower body or other body parts not being detected. When the full upper body is detected, weight measurement is paused to eliminate the influence of the body on the load cell, and the measurement resumes once the upper body moves away from the load cell.
- **Partial Upper Body Detection:** In partial upper body detection, only a part of the upper body is detected within the load cell area. During this time, the load cell continuously reads weight data in real time, and only when the weight change meets the stabilization value is it considered valid. Weight changes less than 0.5 kg are regarded as insignificant fluctuations and are not included in the count. Therefore, when partial upper body detection occurs, the load cell measurement continues, but small weight changes are ignored, and only meaningful changes are reflected in the count.
- **Lower Body Detection:** When the lower body is detected within the load cell area, weight measurement is paused. The weight of the lower body directly affects the load cell, so weight changes are not measured while the lower body is detected. Measurement resumes when the lower body moves away from the load cell area.
- **No Detection:** If the camera does not detect any part of the body over the load cell, the load cell continuously measures weight changes in real time and counts the product based on the weight variations. In the no detection state, the load cell counting process proceeds normally, and the measured weight changes are used to calculate the product count.

3.2.2. Overall Flowchart of the Product Counting Algorithm

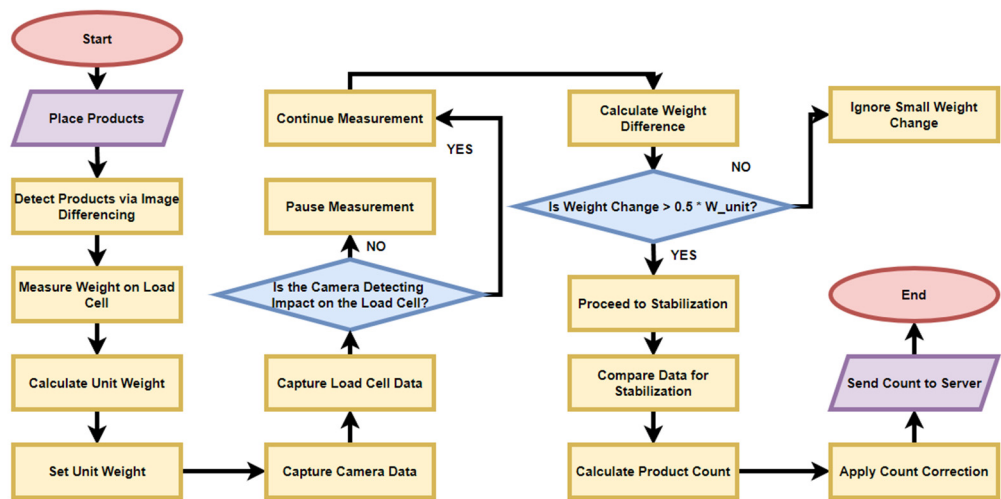


Figure 9. This figure shows the step-by-step process flow of a product counting system.

3.2.3. Overall Flowchart of the Product Counting Algorithm

The load cell used in this system can measure weights up to 2000kg and provides a resolution of 500g. While higher resolution increases the precision of the load cell system, it also raises the cost. Load cells with high resolution enable precise measurements, but in many cases, the level of precision

exceeds what is required in industrial settings. In particular, the products measured in this system mostly weigh over 1kg, so a resolution of 500g is sufficient for accurate weight counting. This study focuses on developing a method to count products using weight increments of 500g, based on these conditions.

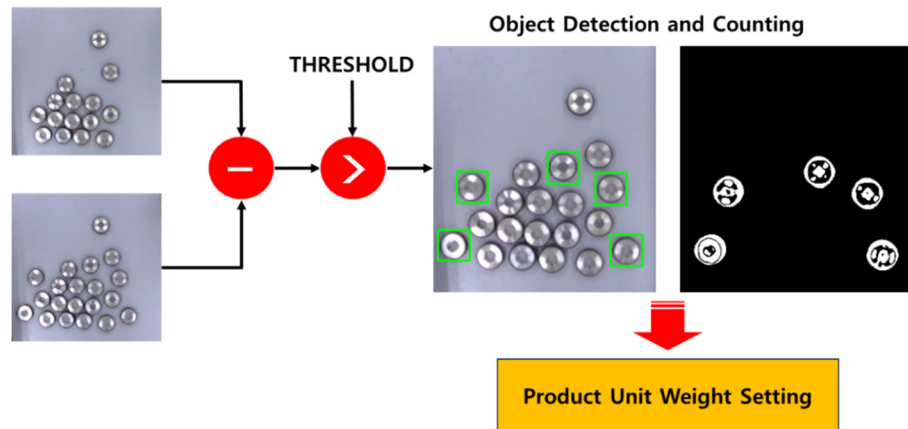


Figure 10. This figure shows the process of unit weight calculation using the image differencing technique.

In this system, the image differencing technique is used to determine the unit weight of products [47–49]. The image differencing method detects the number of products placed on the load cell and calculates the unit weight based on the total weight of the products. During the initial setup, a certain number of products are placed on the load cell, and the unit weight is calculated by using the number of products detected through image differencing and the total weight measured by the load cell. This technique enables accurate detection of the number of products, minimizing weight measurement errors caused by product placement. To enhance the reliability of the image differencing method, adjustments were made to compensate for external factors such as camera angle, lighting, and background noise. This ensures minimal impact from environmental changes on product detection, improving the accuracy of the product counting process.

$$W_{unit} = \frac{W_{Total}}{N_{product}} \quad (10)$$

Here, W_{unit} represents the unit weight of the product, W_{Total} is the total weight measured by the load cell, and $N_{product}$ is the number of products detected using the image differencing technique. Once the initial unit weight is established, the number of products is calculated based on the change in weight measured by the load cell, following these steps:

- **Weight Change Calculation:** When products are added or removed, the weight change is calculated by determining the difference between the current and previous weights measured by the load cell. The weight change must exceed a certain percentage of the unit weight (e.g., 0.5) to be considered valid. This prevents counting errors caused by minor weight fluctuations.

$$\Delta W = W_{current} - W_{previous} \quad (11)$$

- **Stabilization Process:** To improve counting accuracy, the system detects the point at which the weight change stabilizes. Based on the number of data points received per second, N_{DATA} , if the same weight change is detected over a certain number of consecutive readings, the weight is considered stabilized, and the counting is performed. This helps to reduce errors caused by temporary weight fluctuations or noise.
- **Precise Decimal Handling:** The weight change is calculated with precision down to the decimal places, and rounding or truncation is applied only at the time of counting. This minimizes counting errors that may occur when multiple products are loaded simultaneously. For example, if the weight change appears in increments of 0.5, it is rounded up and reflected in the final count.
- **Counting Execution:** Once the stabilization process is complete, the number of products is calculated by dividing the weight change by the unit weight. During this process, decimal values

are carefully handled, and if the first decimal place is 0.5, rounding up or down is applied to ensure the accuracy of the count.

- **Stabilization Process:** To enhance counting accuracy, the system detects when the weight change stabilizes. Based on the number of data points received per second, N_{DATA} , if the same weight change is detected consistently over a certain number of readings, the weight is considered stabilized, and counting is performed. This helps reduce errors caused by temporary weight fluctuations or noise. Here, $C_{product}$ represents the number of products, ΔW is the weight change, and W_{unit} is the unit weight.

$$C_{product} = \frac{\Delta W}{W_{unit}} \quad (12)$$

- **Count Error Correction:** In this system, only products weighing 1kg or more are counted. If the weight change is less than 1kg, it is excluded from the count. Specifically, weight changes of 0.5kg or less are considered minor fluctuations and are disregarded in the count results. This prevents errors caused by small weight changes and ensures that only the actual weight changes of the products are accurately reflected in the count.



Figure 11. This is an image of the test result showing the product counting accuracy.

Table 2. Product Counting Accuracy Test Results.

Test	Product	Product Weight	Actual Quantity	Estimated Quantity	Accuracy(%)
1	A	6.00	184	183	99.46
2	B	43.58	24	25	95.83
3	C	3.1	254	254	100
4	D	3.93	92	91	98.91
5	E	11.80	42	42	100
6	F	4.27	66	67	98.48
7	G	14.44	35	35	100
8	H	4.23	111	111	100
9	I	1.35	50	50	100
10	J	2.3	30	30	100

$$\text{Overall Accuracy(\%)} = \frac{\sum \text{Accuracy for Each Test}}{\text{Number of Tests}} \quad (13)$$

The experiment was conducted in collaboration with a manufacturer of automotive parts, and the test involved counting 10 products ranging from 1kg to 45kg. An accuracy of 99.268% was achieved in the product counting test.

4. Discussion

This study proposes a robust image comparison method that combines traditional image similarity metrics such as SSIM, PSNR, and MSE with the SIFT algorithm to quantitatively evaluate differences between normal and defective products in the manufacturing process. Conventional metrics like SSIM, PSNR, and MSE are insensitive to rotation and positional changes, making it challenging to accurately detect defects when products are captured from various angles. Specifically, for circular products or those with important surface patterns, when the product is photographed in a rotated state, the image similarity decreases, increasing the likelihood of misinterpreting a similar product as defective. To overcome this limitation, the SIFT algorithm was employed to correct for rotation and scale changes, allowing for more reliable image comparisons. The SIFT algorithm detects key points within the image, calculates the orientation and scale of each point, and generates descriptors invariant to these changes. This process enables images to be restored to the same orientation, even when the product is rotated, significantly enhancing the reliability of SSIM, PSNR, and MSE similarity metrics. For products such as wheels or those with critical surface patterns, the accuracy of similarity evaluations improved significantly after applying SIFT for angle restoration. Furthermore, to compensate for the limitations of SSIM, PSNR, and MSE, a weighted combination score, $S_{combined}$, was introduced. This score reflects the characteristics of each metric, allowing for a comprehensive evaluation of both the overall structure and fine differences of the product. The experimental results demonstrated that $S_{combined}$ enabled more precise defect detection than using SSIM, PSNR, or MSE alone, and adjusting the weights for specific defect types allowed for flexible adaptation to various defect scenarios.

Additionally, the product counting system developed in this study applied the YOLO v8 pose algorithm to effectively reduce counting errors caused when a person steps on the load cell. By detecting the presence of a person through body detection, the system automatically paused weight measurement when human influence was detected and resumed measurement when nobody was detected, ensuring accurate product counting. To overcome the resolution limitations of the load cell, the system used an image differencing technique to determine the unit weight of the product, dividing the total weight by the number of detected products to calculate an accurate count. This method involved loading a predetermined number of products onto the load cell, measuring the total weight, and calculating the unit weight based on the number of products detected using the image differencing technique. Using this calculated unit weight, subsequent product counts were derived by dividing the total measured weight by the unit weight. A key aspect of this process was detecting the point at which weight changes stabilized before collecting data. Since errors are likely to occur if weight measurements are not stabilized, the system included a procedure for collecting data only when the weight had not changed for a period of time. This helped to prevent errors caused by transient weight fluctuations and improved counting accuracy. However, one limitation of this study is that the SIFT algorithm can become computationally intensive in complex environments, which may affect performance in real-time applications. Additionally, SIFT's performance in detecting key points may be influenced by external variables such as lighting changes and background complexity, necessitating the use of complementary algorithms.

5. Conclusions

This study proposes a robust defect detection method that combines the SIFT algorithm with traditional image similarity evaluation techniques such as SSIM, PSNR, and MSE, enabling reliable quality inspection despite product rotation and scale changes. By detecting key points in the image and correcting for rotation and scale using the SIFT algorithm, the system demonstrated the capability for accurate quality control. Additionally, the introduction of the $S_{combined}$ metric allowed for precise defect analysis by leveraging the strengths of SSIM, PSNR, and MSE, offering flexible responses to various defect scenarios. In the load cell-based counting system, the YOLO v8 pose algorithm was employed to correct counting errors in real time when a person was on the load cell. Furthermore, the image differencing technique was used to calculate unit weight, enabling accurate product counting. Experimental results showed a high accuracy of 99.268% for products weighing

between 1kg and 45kg. In conclusion, this research demonstrated that reliable defect detection can be achieved despite rotation and scale changes using the SIFT algorithm, and accurate quality inspection and counting systems can be implemented in manufacturing processes by utilizing the image differencing technique and YOLO v8. This confirmed that high reliability and accuracy can be maintained even in real-time manufacturing environments.

Future research should focus on improving the processing speed of the SIFT algorithm and optimizing the system to maintain stable performance in conditions with lighting changes or complex backgrounds. Particularly, efforts should be made to simplify the algorithm for enhanced real-time performance and to integrate machine learning-based predictive models for advanced automation in defect detection. Additionally, for the load cell-based counting system, it is essential to introduce technologies that enhance resolution or develop methods capable of detecting smaller weight changes with greater precision. Algorithms that can correct counting errors in real-time should be advanced, and the system's stability must be reinforced to remain unaffected by external environmental factors such as temperature and vibration. These advancements are expected to play a crucial role in enhancing the efficiency of quality inspection and counting processes in automated manufacturing systems.

Author Contributions: Conceptualization, C.H.L. and Y.S.K.; methodology, C.H.L. and H.K.K.; software, C.H.L.; validation, C.H.L. and Y.S.K.; formal analysis, C.H.L. and Y.S.K.; investigation, C.H.L. and Y.S.K.; writing—original draft preparation, C.H.L. and Y.S.K.; writing—review and editing, H.K.K.; project administration, C.H.L.; visualization, C.H.L.; supervision, H.K.K.; funding acquisition, H.K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Innovative Human Resource Development for Local Intellectualization program, through the Institute of Information & Communications Technology Planning & Evaluation (IITP), funded by the Korea government (MSIT) under grant number IITP-2024-RS-2023-00259678.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: This work was supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (IITP-2024-RS-2023-00259678)

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Reyes Domínguez, D.; Infante Abreu, M. B.; Parv, A. L. Main Trend Topics on Industry 4.0 in the Manufacturing Sector: A Bibliometric Review. *Appl. Sci.* **2024**, *14* (15), 6450.
2. Wang, Z.; Zhao, L.; Li, H.; Xue, X.; Liu, H. Research on a Metal Surface Defect Detection Algorithm Based on DSL-YOLO. *Sensors (Basel)* **2024**, *24* (19), 6268.
3. Ahmmed, M. S.; Isanaka, S. P.; Liou, F. Promoting Synergies to Improve Manufacturing Efficiency in Industrial Material Processing: A Systematic Review of Industry 4.0 and AI. *Machines* **2024**, *12* (10), 681.
4. Lin, B. H.; Chen, J. C.; Lien, J. J. J. Defect Inspection Using Modified YoloV4 on a Stitched Image of a Spinning Tool. *Sensors* **2023**, *23* (9), 4476.
5. Sara, U.; Akter, M.; Uddin, M. S. Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *J. Comput. Commun.* **2019**, *7* (3), 8–18.
6. Zhou, L.; Zhang, L.; Konz, N. Computer Vision Techniques in Manufacturing. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *53* (1), 105–117.
7. Sabilla, I. A.; Meirisdiana, M.; Sunaryono, D.; Husni, M. Best Ratio Size of Image in Steganography Using Portable Document Format with Evaluation RMSE, PSNR, and SSIM. In Proceedings of the 2021 4th International Conference of Computer and Informatics Engineering (IC2IE); IEEE: September 2021; pp 289–294.
8. Prasannakumar, A.; Mishra, D. Deep Efficient Data Association for Multi-Object Tracking: Augmented with SSIM-Based Ambiguity Elimination. *J. Imaging* **2024**, *10* (7).
9. Shahsavarani, S.; Lopez, F.; Ibarra-Castanedo, C.; Maldague, X. P. Advanced Image Stitching Method for Dual-Sensor Inspection. *Sensors* **2024**, *24* (12), 3778.
10. Zhang, H.; Zheng, R.; Zhang, W.; Shao, J.; Miao, J. An Improved SIFT Underwater Image Stitching Method. *Appl. Sci.* **2023**, *13* (22), 12251.

11. Tsourounis, D.; Kastaniotis, D.; Theoharatos, C.; Kazantzidis, A.; Economou, G. SIFT-CNN: When Convolutional Neural Networks Meet Dense SIFT Descriptors for Image and Sequence Classification. *J. Imaging* 2022, 8 (10), 256.
12. Bansal, M.; Kumar, M.; Kumar, M. 2D Object Recognition: A Comparative Analysis of SIFT, SURF and ORB Feature Descriptors. *Multimed. Tools Appl.* 2021, 80 (12), 18839–18857.
13. Lozano-Vázquez, L. V.; Miura, J.; Rosales-Silva, A. J.; Luviano-Juárez, A.; Mújica-Vargas, D. Analysis of Different Image Enhancement and Feature Extraction Methods. *Mathematics* 2022, 10 (14), 2407.
14. Bellavia, F.; Colombo, C. Is There Anything New to Say About SIFT Matching? *Int. J. Comput. Vis.* **2020**, 128 (7), 1847–1866.
15. Talaie Khoei, T.; Ould Slimane, H.; Kaabouch, N. Deep Learning: Systematic Review, Models, Challenges, and Research Directions. *Neural Comput. Appl.* 2023, 35 (31), 23103–23124.
16. Ghosh, K.; Bellinger, C.; Corizzo, R.; Branco, P.; Krawczyk, B.; Japkowicz, N. The Class Imbalance Problem in Deep Learning. *Mach. Learn.* 2024, 113 (7), 4845–4901.
17. Hütten, N.; Alves Gomes, M.; Hölken, F.; Andricevic, K.; Meyes, R.; Meisen, T. Deep Learning for Automated Visual Inspection in Manufacturing and Maintenance: A Survey of Open-Access Papers. *Appl. Syst. Innov.* **2024**, 7 (1), 11.
18. Zhong, X.; Zhu, J.; Liu, W.; Hu, C.; Deng, Y.; Wu, Z. An Overview of Image Generation of Industrial Surface Defects. *Sensors* **2023**, 23 (19), 8160.
19. Kumar, V.; Lalotra, G. S.; Sasikala, P.; Rajput, D. S.; Kaluri, R.; Lakshmana, K.; Uddin, M. Addressing Binary Classification Over Class Imbalanced Clinical Datasets Using Computationally Intelligent Techniques. In Proceedings of the 2022 Healthcare; MDPI: July 2022; Vol. 10, No. 7, p 1293.
20. Wibowo, A.; Setiawan, J. D.; Afrisal, H.; Mertha, A. A. S. M. M. J.; Santosa, S. P.; Wisnu, K. B.; Caesarendra, W. Optimization of Computational Resources for Real-Time Product Quality Assessment Using Deep Learning and Multiple High Frame Rate Camera Sensors. *Appl. Syst. Innov.* 2023, 6 (1), 25.
21. Archana, R.; Jeevaraj, P. E. Deep Learning Models for Digital Image Processing: A Review. *Artif. Intell. Rev.* **2024**, 57 (1), 11.
22. Burger, W.; Burge, M. J. Scale-Invariant Feature Transform (SIFT). In *Digital Image Processing: An Algorithmic Introduction*; Springer International Publishing: Cham, 2022; pp 709–763.
23. Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, 60, 91–110.
24. Acharya, K. A.; Venkatesh Babu, R.; Vadhiyar, S. S. A Real-Time Implementation of SIFT Using GPU. *J. Real-Time Image Process.* **2018**, 14, 267–277.
25. Karami, E.; Prasad, S.; Shehata, M. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. *arXiv* **2017**, arXiv:1710.02726.
26. Güzel, M. S. A Hybrid Feature Extractor Using Fast Hessian Detector and SIFT. *Technologies* **2015**, 3 (2), 103–110.
27. Kumar, P.; Chauhan, S.; Awasthi, L. K. Human Activity Recognition (HAR) Using Deep Learning: Review, Methodologies, Progress and Future Research Directions. *Arch. Comput. Methods Eng.* 2024, 31 (1), 179–219.
28. Aggarwal, A.; Bhutani, N.; Kapur, R.; Dhand, G.; Sheoran, K. Real-Time Hand Gesture Recognition Using Multiple Deep Learning Architectures. *Signal Image Video Process.* 2023, 17 (8), 3963–3971.
29. Ultralytics. <https://github.com/ultralytics/ultralytics/issues/189> (accessed 2023-11-11).
30. Talaat, F. M.; ZainEldin, H. An Improved Fire Detection Approach Based on YOLO-v8 for Smart Cities. *Neural Comput. Appl.* **2023**, 35 (28), 20939–20954.
31. Ultralytics. YOLOv8—Ultralytics YOLOv8 Documentation. <https://docs.ultralytics.com/models/yolov8/> (accessed 2023-11-11).
32. Terven, J.; Córdova-Esparza, D. M.; Romero-González, J. A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **2023**, 5 (4), 1680–1716.
33. Barton, M.; Budjac, R.; Tanuska, P.; Sladek, I.; Nemeth, M. Advancing Small and Medium-Sized Enterprise Manufacturing: Framework for IoT-Based Data Collection in Industry 4.0 Concept. *Electronics* 2024, 13(13), 2485.
34. Zhen, L.; Li, H. A Literature Review of Smart Warehouse Operations Management. *Front. Eng. Manag.* **2022**, 9 (1), 31–55.
35. Ryalat, M.; Franco, E.; Elmoaqet, H.; Almtireen, N.; Alrefai, G. The Integration of Advanced Mechatronic Systems into Industry 4.0 for Smart Manufacturing. *Sustainability* 2024, 16 (19), 8504.
36. Albayrak Ünal, Ö.; Erkeyman, B.; Usanmaz, B. Applications of Artificial Intelligence in Inventory Management: A Systematic Review of the Literature. *Arch. Comput. Methods Eng.* 2023, 30 (4), 2605–2625.
37. Rocha, J. G.; Couto, C.; Correia, J. H. Smart Load Cells: An Industrial Application. *Sens. Actuators, A* **2000**, 85 (1–3), 262–266.

38. Hossein-Nejad, Z.; Agahi, H.; Mahmoodzadeh, A. Image Matching Based on the Adaptive Redundant Keypoint Elimination Method in the SIFT Algorithm. *Pattern Anal. Appl.* 2021, 24 (2), 669–683. <https://doi.org/10.1007/s10044-020-00880-w>.
39. Hu, X.; Tang, Y.; Zhang, Z. Video Object Matching Based on SIFT Algorithm. In 2008 International Conference on Neural Networks and Signal Processing; IEEE, 2008; pp 412–415. <https://doi.org/10.1109/ICNNSP.2008.4590386>.
40. Alhwarin, F.; Wang, C.; Ristić-Durrant, D.; Gräser, A. Improved SIFT-Features Matching for Object Recognition. In Visions of Computer Science-BCS International Academic Conference; BCS Learning & Development, 2008; pp 165–176.
41. Zhou, H.; Yuan, Y.; Shi, C. Object Tracking Using SIFT Features and Mean Shift. *Comput. Vis. Image Underst.* 2009, 113 (3), 345–352. <https://doi.org/10.1016/j.cviu.2008.08.006>.
42. Setiadi, D. R. I. M. PSNR vs SSIM: Imperceptibility Quality Assessment for Image Steganography. *Multimedia Tools Appl.* 2021, 80 (6), 8423–8444. <https://doi.org/10.1007/s11042-020-10129-4>.
43. Hore, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. In 2010 20th International Conference on Pattern Recognition; IEEE, 2010; pp 2366–2369. <https://doi.org/10.1109/ICPR.2010.579>.
44. Sara, U.; Akter, M.; Uddin, M. S. Image Quality Assessment through FSIM, SSIM, MSE, and PSNR—A Comparative Study. *J. Comput. Commun.* 2019, 7 (3), 8–18. <https://doi.org/10.4236/jcc.2019.73002>.
45. Terven, J.; Córdova-Esparza, D. M.; Romero-González, J. A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* 2023, 5 (4), 1680–1716. <https://doi.org/10.3390/make5040088>.
46. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics; GitHub, 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on January 12, 2023).
47. Piccardi, M. Background Subtraction Techniques: A Review. In 2004 IEEE International Conference on Systems, Man and Cybernetics; IEEE, 2004; Vol. 4, pp 3099–3104. <https://doi.org/10.1109/ICSMC.2004.1400815>.
48. Kalsotra, R.; Arora, S. Background Subtraction for Moving Object Detection: Explorations of Recent Developments and Challenges. *Vis. Comput.* 2022, 38 (12), 4151–4178. <https://doi.org/10.1007/s00371-021-02206-2>.
49. Benezeth, Y.; Jodoin, P. M.; Emile, B.; Laurent, H.; Rosenberger, C. Comparative Study of Background Subtraction Algorithms. *J. Electron. Imaging* 2010, 19 (3), 033003. <https://doi.org/10.1117/1.3454396>.

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.