applied system innovation

Article

Computer-Vision-Based Product Quality Inspection and Novel Counting System

Changhyun Lee , Yunsik Kim and Hunkee Kim *

AI System Lab, Department of Advanced Materials Processing Engineering, Inha University, 36 Gaetbeol-ro, Yeonsu-gu, Incheon 21999, Republic of Korea; ckd9848@gmail.com (C.L.); yskim9701@naver.com (Y.K.)
* Correspondence: hunkeekim@inha.ac.kr

Abstract: In this study, we aimed 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 YOLOv8 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 1 kg, 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; YOLOv8 Pose; load cell; feature matching

check for updates

Citation: Lee, C.; Kim, Y.; Kim, H. Computer-Vision-Based Product Quality Inspection and Novel Counting System. *Appl. Syst. Innov.* **2024**, *7*, 127. https://doi.org/10.3390/asi7060127

Academic Editor: Cheng-Fu Yang

Received: 29 October 2024
Revised: 12 December 2024
Accepted: 13 December 2024
Published: 18 December 2024

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. Using 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 500 g increments, overcounting or undercounting may occur if the actual weight does not 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 YOLOv8 (You Only Look Once version 8) 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 study utilized a high-speed camera with a variable focus function, 1280 × 720 resolution, and a frame rate of 120 fps (frames per second) to maximize efficiency in defect detection and quality control. This high-speed capability ensures precise detection of even minor defects on products moving rapidly on a conveyor belt. The camera's variable focus function allows it to adapt flexibly to various product sizes and shapes, enabling stable defect detection without the need for additional focus adjustments. The camera was installed approximately 50 cm above the conveyor belt and set at a 90-degree angle to capture detailed images of the product surfaces with precision. This configuration provides a standardized setup that can be reliably applied across diverse manufacturing processes. Additionally, the lighting system was designed to optimize illumination and uniformity in the inspection area by supplying light from both directions.

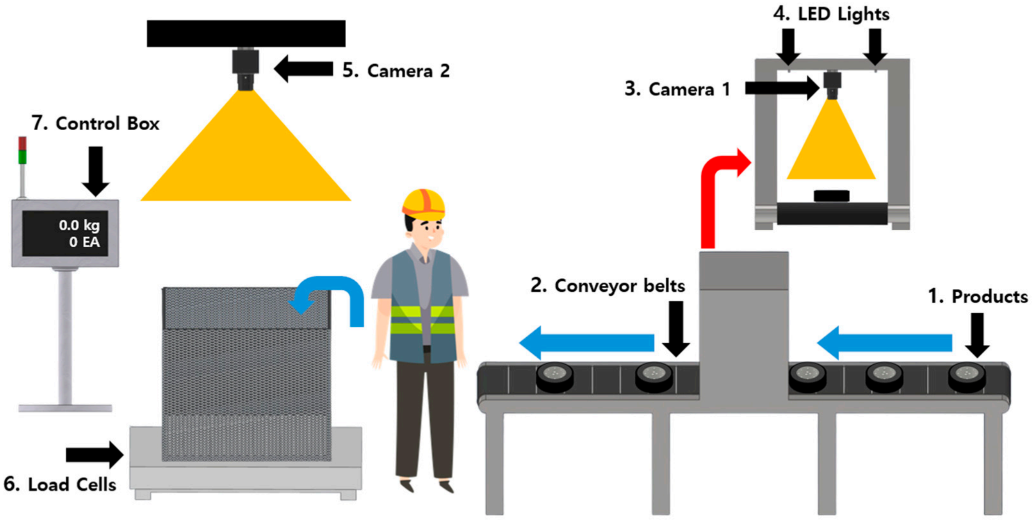

Figure 1. This figure is a schematic image of the entire system.

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 were determined by analyzing the SSIM, PSNR, and MSE values with the $S_{combined}$ formula and difference image technique. Second, we introduced the YOLOv8 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 developed 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 were placed on the load cell and detected through difference images, and then the unit weight is calculated. The product count was subsequently derived based on the total weight. This approach minimized counting errors arising from product placement or external factors.

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.

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, you can either undersample the majority-category data using random sampling techniques to match the minority-category data or use a technique called a generative adversarial network (GAN) to oversample the minority-category data. As shown in Figure 2, the undersampling process (left) involves reducing the number of samples in the majority class (Class A) to match the sample size of the minority class (Class B). This technique helps balance the dataset but may lead to information loss as potentially valuable data from the majority class is discarded. On the other hand, the oversampling process (right) increases the sample size of the minority class by duplicating existing data or generating synthetic samples. This approach helps preserve all the majority-class data while keeping the dataset balanced. Techniques such as generative adversarial networks (GANs) have been widely used to generate realistic synthetic data for the minority class, improving model performance by effectively addressing the class imbalance problem. However, even with these methods, achieving optimal performance remains challenging due to the risk of overfitting in oversampled data and the difficulty of accurately measuring model performance [19].

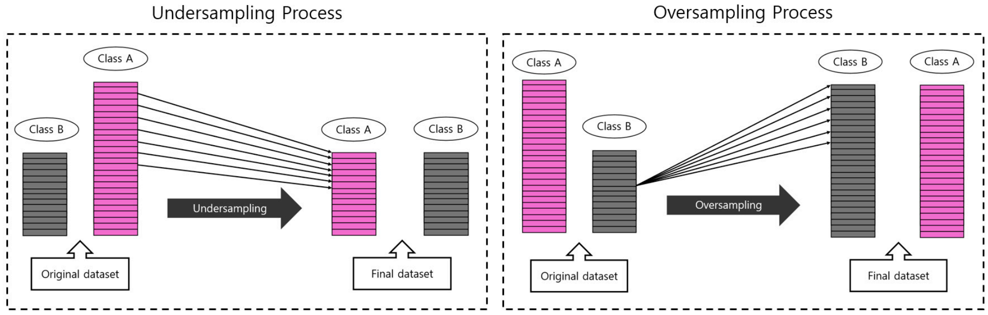

Figure 2. The image on the left shows the undersampling process, which reduces the data of Class A, the majority class, to match the amount of data in Class B, the minority class. On the other hand, the image on the right illustrates the oversampling process, which replicates or generates data from the minority class, Class B, to balance it with the majority class, Class A. These methods are used to address the data imbalance problem.

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 study used computer vision techniques such as SSIM, MSE, PSNR, and SIFT to compare the image data. The goal was 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. These technologies allow for the evaluation of 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.

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 the results in Figure 3, we tested SURF (Speeded-Up Robust Features), SIFT, and the hybrid technique. The hybrid technique demonstrated good performance for scale changes, whereas SIFT provided more accurate performance when detecting features such as rotational invariance [26]. Based on this, this study aimed 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.

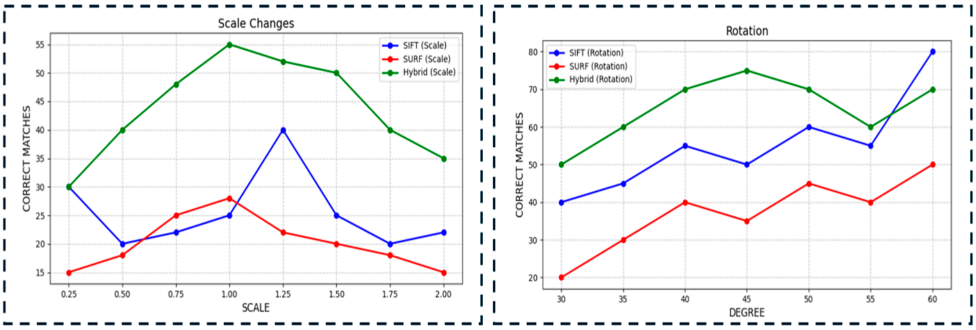

Figure 3. This figure shows a comparative analysis of the performance of the SIFT, SURF, and hybrid algorithms. On the left is the performance metric for scale change, and on the right is the performance metric for rotation.

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

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 can indicate which approaches may be more effective in particular situations by comparing different architectures.

Figure 4 provides an overview of the YOLOv8 Pose architecture, which enhances the standard YOLOv8 model with advanced pose estimation capabilities [29,30]. The architecture consists of three main components: the backbone, neck, and head. The backbone performs feature extraction using convolutional layers and advanced modules such as C2f and SPPF, ensuring efficient and robust feature representation. The neck fuses features from multiple scales using upsampling and concatenation layers, effectively integrating spatial and semantic information. The head is adapted for pose estimation, providing dual outputs for bounding box coordinates and keypoint locations, enabling simultaneous object detection and pose estimation. Unlike earlier YOLO models, YOLOv8 Pose adopts an anchor-free design, simplifying the detection process and improving computational efficiency [31,32]. With enhanced speed and accuracy, the architecture is particularly suitable for real-time applications such as human motion tracking and activity recognition. The integration of the C2f module for lightweight computation and the optimized SPPF module for global feature extraction further contribute to its high performance. In summary, YOLOv8 Pose combines innovative features and real-time capabilities, making it a versatile tool for pose estimation and activity recognition in challenging environments.

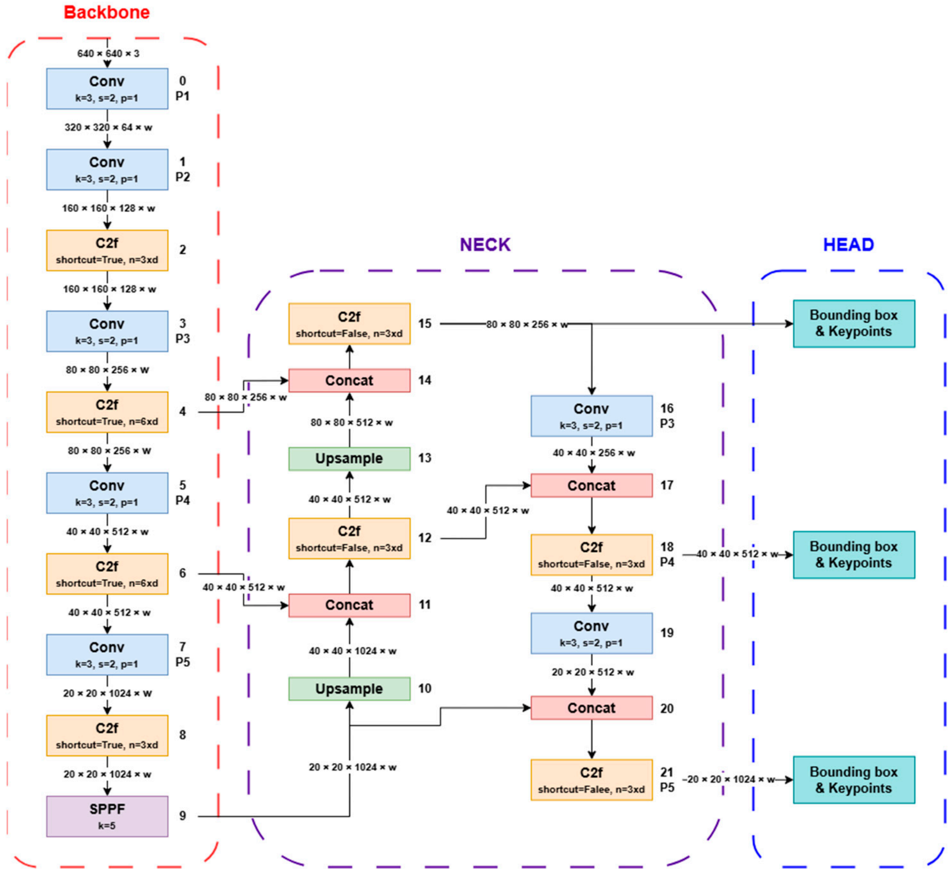

Figure 4. This figure illustrates the architecture of YOLOv8 Pose, highlighting its backbone, neck, and head modules with additional functionality for pose estimation.

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.

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 500 g, overcounting or undercounting 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 study, the goal is to develop a system that improves the accuracy of product counting and inventory management processes by setting the product counting value as the unit weight. This is achieved using a technology that recognizes when a worker approaches the load cell and prevents counting from being affected, along with the difference image technique. We believe that this system will contribute to reducing product counting errors and increasing the reliability of inventory management by superseding the limitations of existing techniques.

3. Methods and Result

This study aimed 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. In particular, 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 applied 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 the 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 5, 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 [39]:

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

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:

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

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

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:

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

Here, p_i and q_i represent the iii-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:

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

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 determination of whether the image has been scaled up or down.

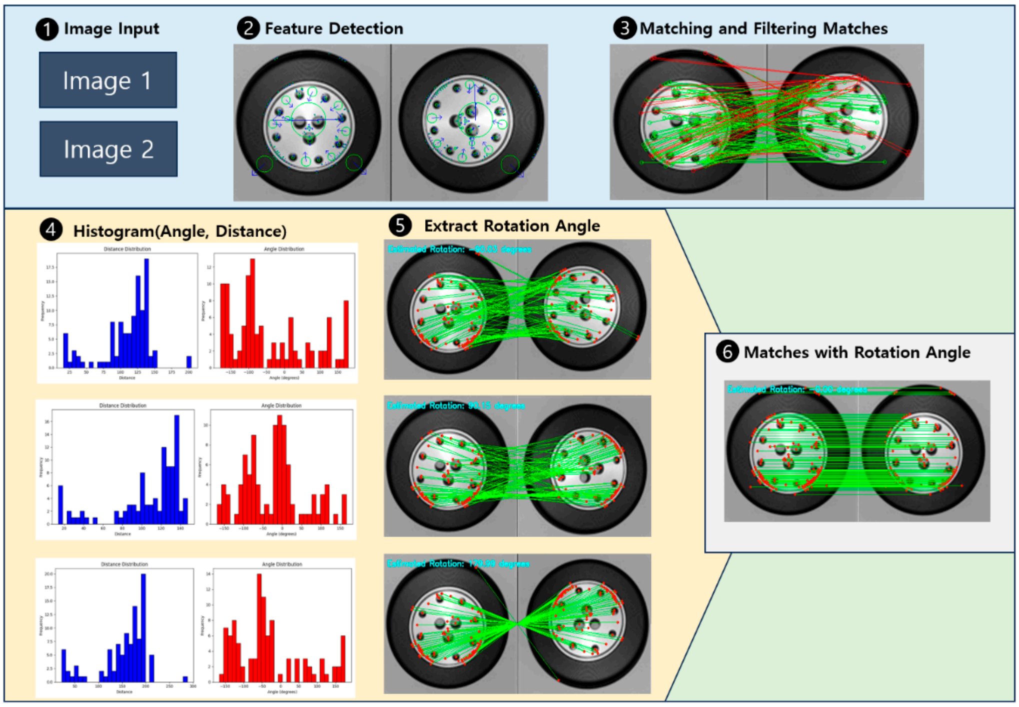
1 Image Input
Image 1
Image 2
2 Feature Detection
3 Matching and Filtering Matches
4 Histogram(Angle, Distance)
Distance Distribution
Angle Distribution
Frequency
Distance
Angle (degrees)
5 Extract Rotation Angle
Estimated Rotation: −90.03 degrees
Estimated Rotation: 90.15 degrees
Estimated Rotation: 179.99 degrees
6 Matches with Rotation Angle
Estimated Rotation: −0.00 degrees

Figure 5. This figure illustrates the process of image alignment and analysis using the SIFT algorithm. The blue arrows represent the orientation of the detected keypoints based on intensity gradients, while the green circles and dots indicate the location and scale of the keypoints. The green lines connect the matched keypoints between two images, enabling visualization of their correspondence for rotation and alignment. The histograms in the figure provide a quantitative analysis of the matched keypoints, where the red histograms represent the angle distribution, and the blue histograms represent the distance distribution, offering insights into the rotational and geometric transformation between the two images.

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

Figure 6 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, PSNR, and MSE were applied. These metrics were 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, and the formulas for each method shown below in Table 1 [42,43].

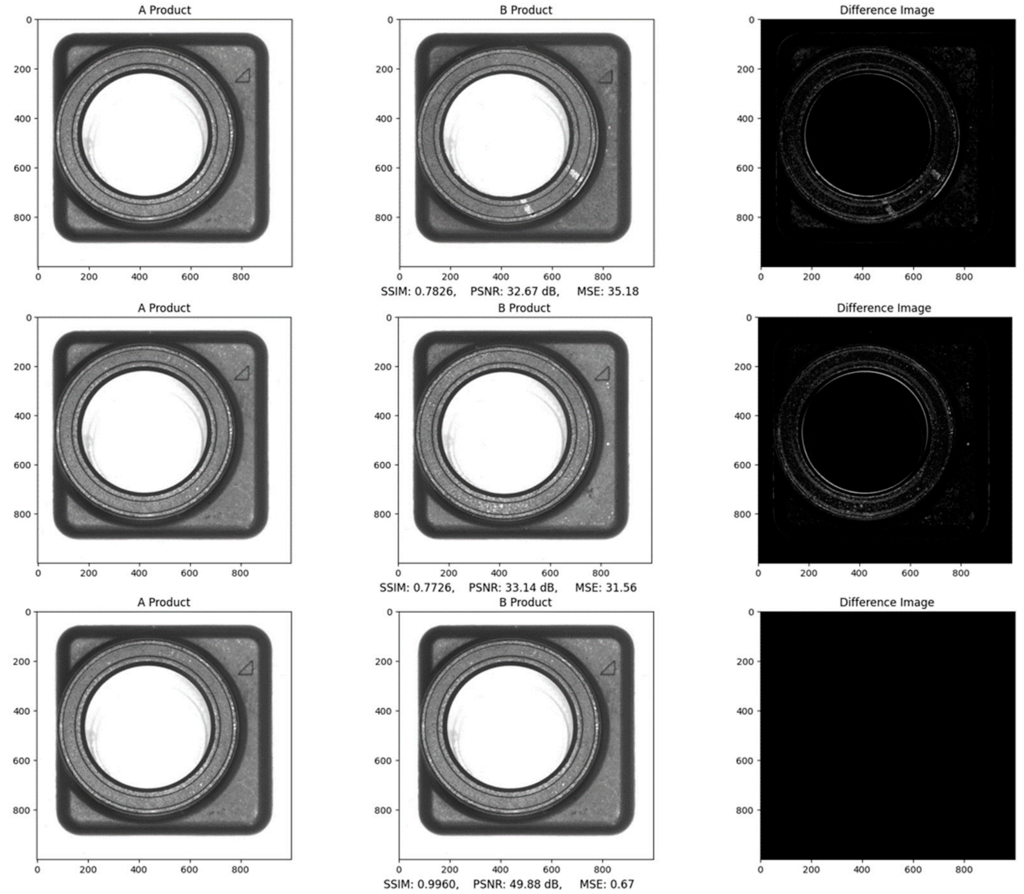
A Product
B Product
Difference Image
0
200
400
600
800
SSIM: 0.7826, PSNR: 32.67 dB, MSE: 35.18
SSIM: 0.7726, PSNR: 33.14 dB, MSE: 31.56
SSIM: 0.9960, PSNR: 49.88 dB, MSE: 0.67

Figure 6. 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	$\frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}$ (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	$\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[I_1(i,j)-I_2(i,j)]^2$ (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	$10\log_{10}\left(\frac{MAX_I^2}{MSE}\right)$ (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:

$$S_{\text{combined}} = \omega_{\text{SSIM}} * S_{\text{SSIM}} + \omega_{\text{PSNR}} * S_{\text{PSNR}} + \omega_{\text{MSE}} * \frac{1}{S_{\text{MSE}}} \tag{9}$$

The weights ω_{SSIM}, ω_{PSNR}, and 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 the 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

In this study, we developed a product counting system that applies the YOLOv8 Pose algorithm to detect the human body and correct errors that occur when a person steps onto the load cell [44,45]. 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. As shown in Figure 7, the control images illustrate the situations of full upper body detection, partial upper body detection, lower body detection, and no detection.

- 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.

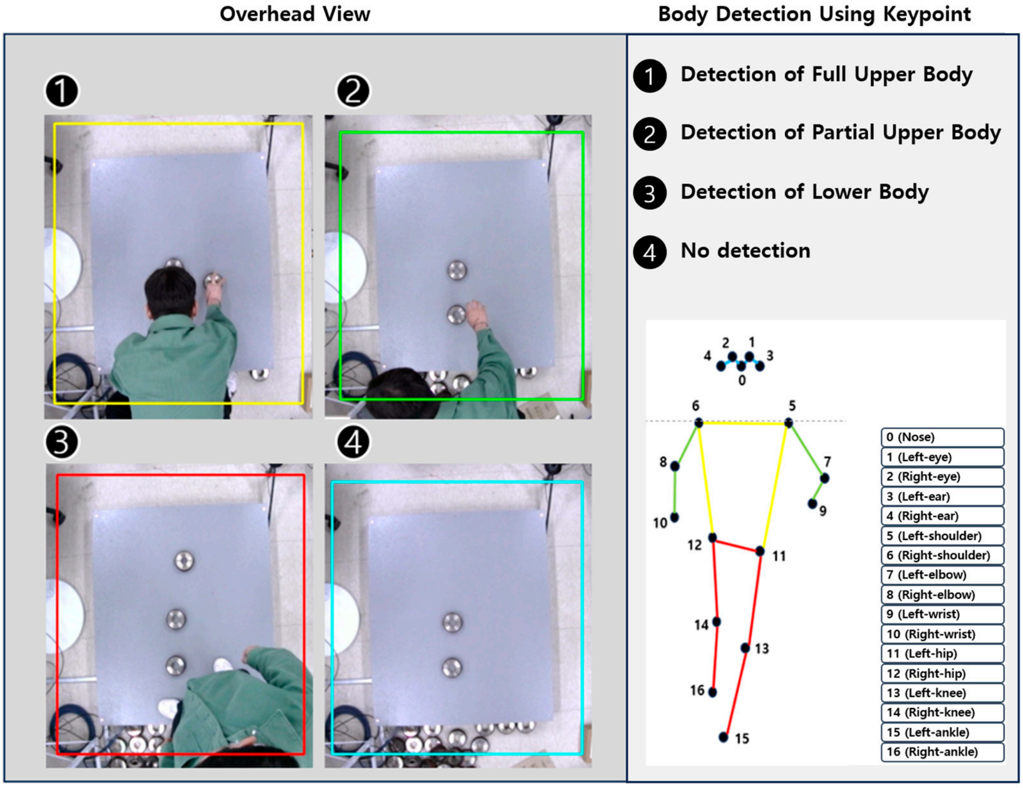

Figure 7. This figure shows the case-based operation in the load cell counting system using a body detection algorithm.

3.2.2. Overall Flowchart of the Product Counting Algorithm

The load cell used in the product counting system for Figure 8 can weigh up to 2000 kg and has a resolution of 500 g. 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 1 kg, so a resolution of 500 g is sufficient for accurate weight counting. This study focuses on developing a method to count products using weight increments of 500 g, based on these conditions.

3.2.3. Unit Weight Calculation and Counting Method Using Image Differencing Technique

As presented in Figure 9, image difference techniques are employed to determine the unit weight of a product [46–48]. 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.

$$W_{\text{unit}} = \frac{W_{\text{Total}}}{N_{\text{product}}} \tag{10}$$

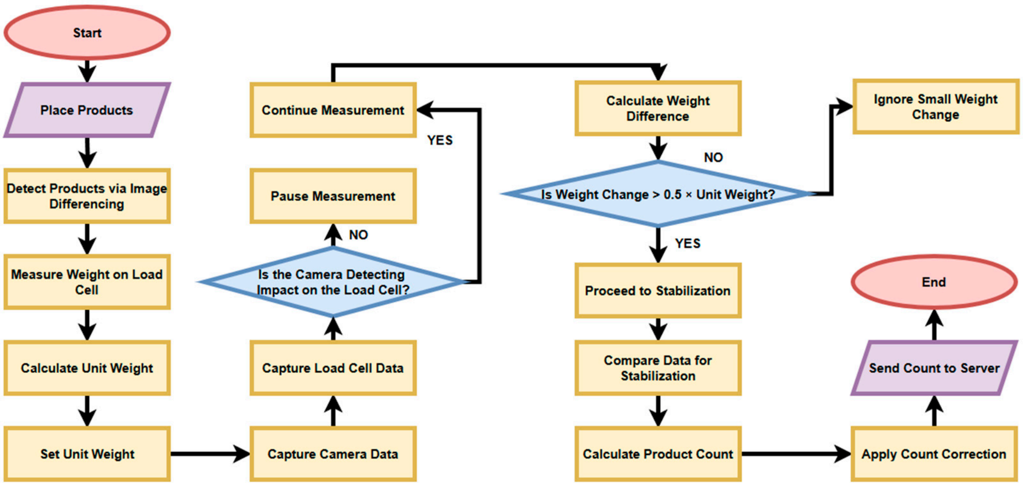

Figure 8. This figure shows the step-by-step process flow of a product counting system.

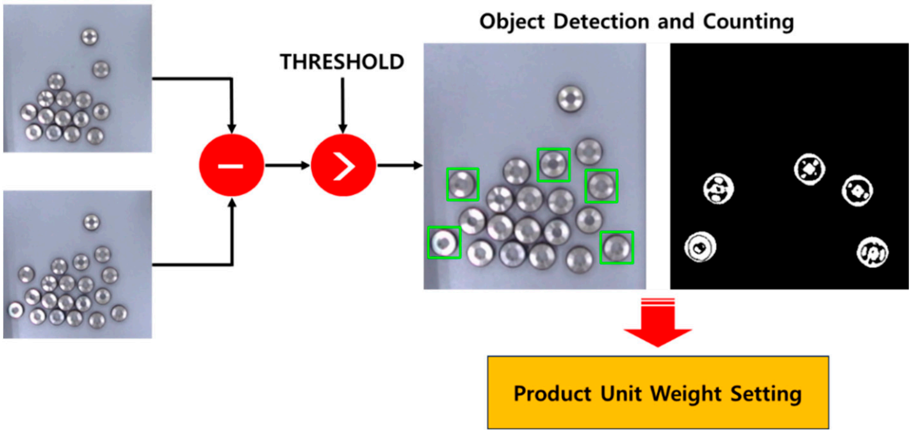

Figure 9. This figure shows the process of unit weight calculation using the image differencing technique.

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.

$$\Delta W = W_{\text{current}} - W_{\text{previous}} \tag{11}$$

- 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 to the first 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.

$$C_{product} = \frac{\Delta W}{W_{unit}} \tag{12}$$

- Count Error Correction: In this system, only products weighing 1 kg or more are counted. If the weight change is less than 1 kg, it is excluded from the count. Specifically, weight changes of 0.5 kg 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.

The experimental results presented in Figure 10 and Table 2 were obtained in collaboration with an automobile parts manufacturer to verify product accuracy. The test was conducted by counting 10 products ranging in weight from 1 kg to 45 kg, achieving an accuracy of 99.268% in the product coefficient test.

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

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	3.10	254	254	100.00
4	D	3.93	92	91	98.91
5	E	11.80	42	42	100.00
6	F	4.27	66	67	98.48
7	G	14.44	35	35	100.00
8	H	4.23	111	111	100.00
9	I	1.35	50	50	100.00
10	J	2.30	30	30	100.00

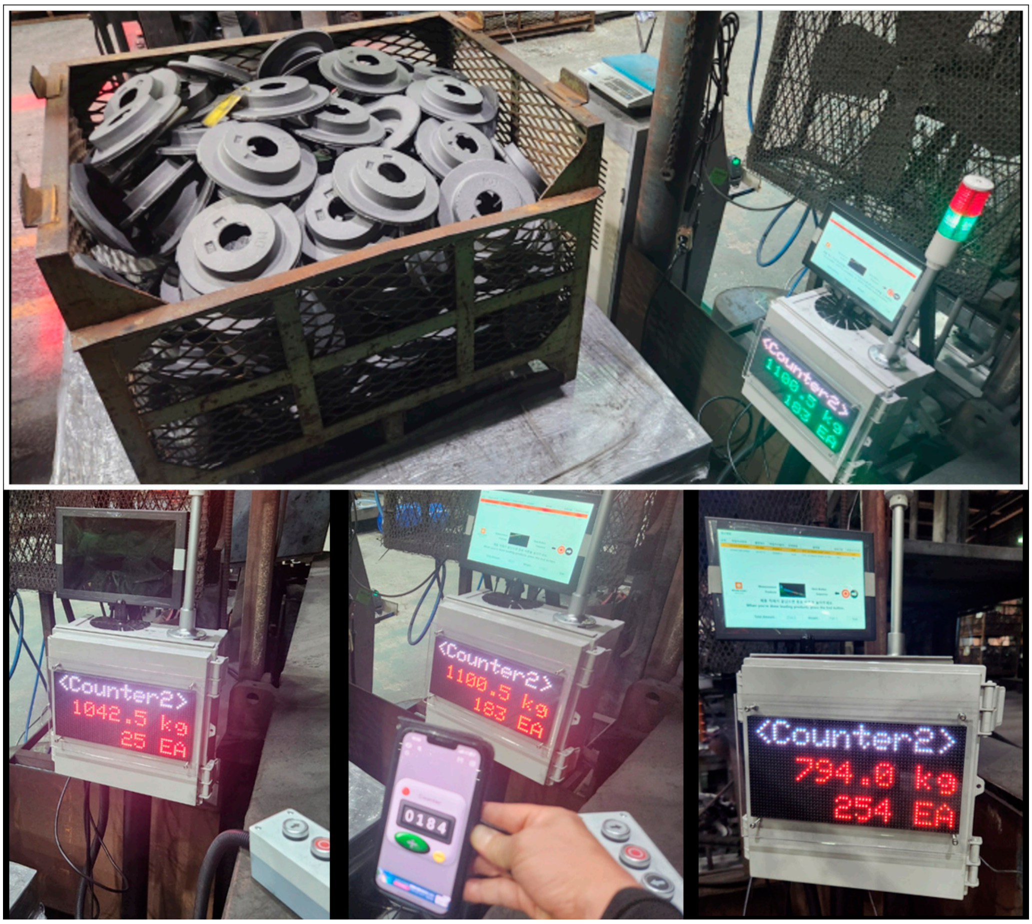

Figure 10. Images of the test result showing the product counting accuracy.

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 accurate and dependable 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 YOLOv8 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 YOLOv8 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 1 kg and 45 kg. 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 YOLOv8. 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. In particular, 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.L. and Y.K.; methodology, C.L. and H.K.; software, C.L.; validation, C.L. and Y.K.; formal analysis, C.L. and Y.K.; investigation, C.L. and Y.K.; writing—original draft preparation, C.L. and Y.K.; writing—review and editing, H.K.; project administration, C.L.; visualization, C.L.; supervision, H.K.; funding acquisition, H.K. All authors have read and agreed to the published version of the manuscript.

Funding: 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).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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*, 6450. [CrossRef]
2. Wang, Z.; Zhao, L.; Li, H.; Xue, X.; Liu, H. Research on a Metal Surface Defect Detection Algorithm Based on DSL-YOLO. *Sensors* **2024**, *24*, 6268. [CrossRef]
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*, 681. [CrossRef]
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*, 4476. [CrossRef] [PubMed]
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*, 8–18. [CrossRef]
6. Zhou, L.; Zhang, L.; Konz, N. Computer Vision Techniques in Manufacturing. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *53*, 105–117. [CrossRef]
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), Depok, Indonesia, 14–15 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*, 171. [CrossRef]
9. Shahsavarani, S.; Lopez, F.; Ibarra-Castanedo, C.; Maldague, X.P. Advanced Image Stitching Method for Dual-Sensor Inspection. *Sensors* **2024**, *24*, 3778. [CrossRef]
10. Zhang, H.; Zheng, R.; Zhang, W.; Shao, J.; Miao, J. An Improved SIFT Underwater Image Stitching Method. *Appl. Sci.* **2023**, *13*, 12251. [CrossRef]
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*, 256. [CrossRef]
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*, 18839–18857. [CrossRef]
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*, 2407. [CrossRef]
14. Bellavia, F.; Colombo, C. Is There Anything New to Say About SIFT Matching? *Int. J. Comput. Vis.* **2020**, *128*, 1847–1866. [CrossRef]
15. Talaei Khoei, T.; Ould Slimane, H.; Kaabouch, N. Deep Learning: Systematic Review, Models, Challenges, and Research Directions. *Neural Comput. Appl.* **2023**, *35*, 23103–23124. [CrossRef]
16. Ghosh, K.; Bellinger, C.; Corizzo, R.; Branco, P.; Krawczyk, B.; Japkowicz, N. The Class Imbalance Problem in Deep Learning. *Mach. Learn.* **2024**, *113*, 4845–4901. [CrossRef]
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*, 11. [CrossRef]
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*, 8160. [CrossRef] [PubMed]
19. Kumar, V.; Lalotra, G.S.; Sasikala, P.; Rajput, D.S.; Kaluri, R.; Lakshmanna, K.; Uddin, M. Addressing Binary Classification Over Class Imbalanced Clinical Datasets Using Computationally Intelligent Techniques. *Healthcare* **2022**, *10*, 1293. [CrossRef] [PubMed]
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*, 25. [CrossRef]
21. Archana, R.; Jeevaraj, P.E. Deep Learning Models for Digital Image Processing: A Review. *Artif. Intell. Rev.* **2024**, *57*, 11. [CrossRef]

22. Burger, W.; Burge, M.J. Scale-Invariant Feature Transform (SIFT). In *Digital Image Processing: An Algorithmic Introduction*; Springer International Publishing: Cham, Switzerland, 2022; pp. 709–763.
23. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
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. [CrossRef]
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*, 103–110. [CrossRef]
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*, 179–219. [CrossRef]
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*, 3963–3971. [CrossRef]
29. Maji, D.; Nagori, S.; Mathew, M.; Poddar, D. Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 2637–2646.
30. Ultralytics. YOLOv8—Ultralytics YOLOv8 Documentation. Available online: https://docs.ultralytics.com/models/yolov8/ (accessed on 11 December 2023).
31. Dong, C.; Du, G. An Enhanced Real-Time Human Pose Estimation Method Based on Modified YOLOv8 Framework. *Sci. Rep.* **2024**, *14*, 8012. [CrossRef]
32. Dong, C.; Tang, Y.; Zhang, L. HDA-Pose: A Real-Time 2D Human Pose Estimation Method Based on Modified YOLOv8. *Signal Image Video Process.* **2024**, *18*, 5823–5839. [CrossRef]
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*, 2485. [CrossRef]
34. Zhen, L.; Li, H. A Literature Review of Smart Warehouse Operations Management. *Front. Eng. Manag.* **2022**, *9*, 31–55. [CrossRef]
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*, 8504. [CrossRef]
36. Albayrak Ünal, Ö.; Erkayman, B.; Usanmaz, B. Applications of Artificial Intelligence in Inventory Management: A Systematic Review of the Literature. *Arch. Comput. Methods Eng.* **2023**, *30*, 2605–2625. [CrossRef]
37. Rocha, J.G.; Couto, C.; Correia, J.H. Smart Load Cells: An Industrial Application. *Sens. Actuators A* **2000**, *85*, 262–266. [CrossRef]
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*, 669–683. [CrossRef]
39. Zhou, H.; Yuan, Y.; Shi, C. Object Tracking Using SIFT Features and Mean Shift. *Comput. Vis. Image Underst.* **2009**, *113*, 345–352. [CrossRef]
40. Alhwarin, F.; Wang, C.; Ristić-Durrant, D.; Gräser, A. Improved SIFT-Features Matching for Object Recognition. In Proceedings of the Visions of Computer Science-BCS International Academic Conference, London, UK, 22–24 September 2008; BCS Learning & Development. pp. 165–176.
41. Hu, X.; Tang, Y.; Zhang, Z. Video Object Matching Based on SIFT Algorithm. In Proceedings of the 2008 International Conference on Neural Networks and Signal Processing, Zhenjiang, China, 7–11 June 2008; pp. 412–415. [CrossRef]
42. Setiadi, D.R.I.M. PSNR vs SSIM: Imperceptibility Quality Assessment for Image Steganography. *Multimedia Tools Appl.* **2021**, *80*, 8423–8444. [CrossRef]
43. Hore, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369. [CrossRef]
44. 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*, 1680–1716. [CrossRef]
45. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics; GitHub. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 12 January 2023).
46. Piccardi, M. Background Subtraction Techniques: A Review. In Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics, The Hague, The Netherlands, 10–13 October 2004; Volume 4, pp. 3099–3104. [CrossRef]
47. Kalsotra, R.; Arora, S. Background Subtraction for Moving Object Detection: Explorations of Recent Developments and Challenges. *Vis. Comput.* **2022**, *38*, 4151–4178. [CrossRef]
48. Benezeth, Y.; Jodoin, P.M.; Emile, B.; Laurent, H.; Rosenberger, C. Comparative Study of Background Subtraction Algorithms. *J. Electron. Imaging* **2010**, *19*, 033003. [CrossRef]

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.