sustainability

MDPI

Article

Deep Learning-Based Defect Detection for Sustainable Smart Manufacturing

Sang-Hyun Park [1], Kang-Hee Lee [1], Ji-Su Park [2] and Youn-Soon Shin [1,*]

[1] Department of Computer Science and Engineering, Dongguk University, Seoul 04620, Korea; powerful104@dgu.ac.kr (S.-H.P.); dlrkdgml1998@dgu.ac.kr (K.-H.L.)
[2] Department of Computer Science and Engineering, Jeonju University, Jeonju 55069, Korea; jisupark@jj.ac.kr
* Correspondence: ysshin@dgu.edu

Abstract: In manufacturing a product, product defects occur at several stages. This study makes the case that one can build a smart factory by introducing it into the manufacturing process of small-scale scarce products, which mainly solves the defect problem through visual inspection. By introducing an intelligent manufacturing process, defects can be minimized, and human costs can be lowered to enable sustainable growth. In this paper, in order to easily detect defects occurring in the manufacturing process, we studied a deep learning-based automatic defect detection model that can train product characteristics and determine defects using open sources. To verify the performance of this model, it was applied to the disposable gas lighter manufacturing process to detect the liquefied gas volume defect of the lighter, and it was confirmed that the detection accuracy and processing time were sufficient to apply to the manufacturing process.

Keywords: deep learning; image processing; smart factory; sustainable computing; Internet of Things

check for updates

Citation: Park, S.-H.; Lee, K.-H.; Park, J.-S.; Shin, Y.-S. Deep Learning-Based Defect Detection for Sustainable Smart Manufacturing. *Sustainability* **2022**, *14*, 2697. https://doi.org/10.3390/su14052697

Academic Editors: Arun Kumar Sangaiah, Xi Zheng, Changqing Luo, Shuihua Wang and Ankit Chaudhary

Received: 16 January 2022
Accepted: 24 February 2022
Published: 25 February 2022

1. Introduction

Recently, many companies have promoted the smart factorization of their processes to increase sales and process efficiency [1]. In the case of large companies, it is relatively easy to establish an intelligent factory system in the manufacturing process based on enormous capital and technology. It is costly to introduce a complete solution for defect detection to industrial sites with little or no background knowledge on image processing or artificial intelligence algorithms. Therefore, we propose a low-cost module-type solution using a single-board computer that can replace expensive solutions to achieve a greater effect through low-cost investment. Although there is an initial installation cost, it can be expected to gradually reduce labor costs through automation. By applying the proposed solution to some factories with a small-scale mass production system as a specific product among SMEs, it helps the transition to smartization.

The proposed model can detect errors in real time using a general, easy-to-correct, and easy-to-apply error detection model rather than a model specific to distinct products and processes. In addition, the applicability of the proposed model was verified by applying it to the domestic disposable gas lighter manufacturing process, which is currently promoting smart factories. This study aims to prove that the model proposed by our research team can contribute to the manufacturing process of SMEs by applying it to the operation of automatically detecting defects in the volume of liquefied gas injected into the disposable gas lighter.

This study developed an automatic defect detection model based on the YOLOv4 (You Only Look Once) deep learning object recognition model, OpenCV (Open-source Computer Vision) real-time image processing, and a single-board computer for IoT (Internet of Things). This model can be easily applied to the manufacturing process even in SMEs with low capital and technology. The proposed model pre-processes the image of the product produced in the process so that it is not affected by the factory environment. Then, it can successfully detect errors by using the product defect standard set in advance [2].

2. Related Work

2.1. Smart Factory Defect Detection System

In the manufacturing process of the actual product, various factors may cause multiple defects to the extent that the product cannot be released in the market, so it is essential to continuously monitor and remove defective products in advance. From this point of view, several studies have been conducted to monitor the manufacturing process efficiently and accurately.

In Ref. [3], a two-stage defect detection network based on the object detection model YOLO and the normalizing flow-based defect detection model Different is presented. Additionally, multi-scale image transformations are used to solve the issue of product images cropped by YOLO, including too much background noise. They obtain 99.2% AUROC (Area Under Receiver Operating Characteristic) by cropping the top and bottom by 10% and the left and right by 5%.

In Ref. [4], YOLO using the CNN (Convolutional Neural Network) is applied with some modifications to detect defects with complex and unique features, such as defects on steel surfaces. The system was mounted on a chipset and placed on the factory line to provide the size and location of the defect, significantly improving the detection process. Despite the small data set, the system achieved 70.66% mAP (mean Average Precision).

In Ref. [5], the YOLOv3 network model was used to solve insensitivity to defect targets and the lack of real-time detection in bearing factories' existing defect detection algorithm. The model was divided into four sub-models: the bottleneck attention network (BNA-Net), the attention prediction subnet model, the defect localization subnet model, and the large-size output feature branch. As a result of the experiment using the model, the obtained mAP was 69.74%.

In Ref. [6], deep learning is used for an automatic classification system for machine parts. The proposed method is combined with a robotic arm to process mechanical part classification efficiently. Object recognition is performed using a region-based full convolutional network (R-FCN) trained with the pre-labeled image, and the object is recognized from the image captured by the camera. It is possible to identify even vertically and horizontally rotated images of objects with enough data.

In Ref. [7], the methodology and development problems of several widely used R-CNN models and YOLO models are analyzed, and they propose a real-time image recognition model and architecture suitable for integrated circuit boards (ICBs) in the manufacturing process. Various types of ICBs were collected and used as model training datasets, and a preliminary image recognition model was constructed to classify and predict them based on different features of each ICB, and the proposed model showed an average accuracy of 96.53%. This suggests that accurate and immediate image recognition in the smart manufacturing process can reduce staffing, improve equipment efficiency, and increase productivity and efficiency in the production line.

In Ref. [8], an automated computer vision pipeline is proposed for object recognition. In this method, data are augmented to balance the object classes, and then the CNN is modified according to the selected data set. The revised model is trained by transfer learning and extract features. The extracted features include some redundant information removed using an improved whale optimization algorithm (WOA). Finally, features are classified using several supervised learning algorithms for final recognition. The method achieved an accuracy of 93% when performed using the dataset from Caltech101.

In Ref. [9], a detection algorithm for small parts based on deep learning is proposed. They select the requirements for an intelligent production process to obtain a reliable and accurate real-time micro-part defect detection system. The algorithm proposed in this paper has a higher defect prediction probability than YOLOv3, Faster-RCNN, and FPN. It was highly accurate and had the lowest probability of missing a defect. A correlation model was established between the detection capability of the system for detecting component defects and the speed of conveyor movement. The defect detection accuracy was highest when the speed of the conveyor belt was 7.67 m/min.

In Ref. [10], a deep convolutional neural network is used in the visual inspection process to classify the bearing mounting into two classes: good or bad. Everything in this model is open source, as the goal of this research is to prove the viability for smaller companies to use AI in their everyday processes, in order to improve the overall business efficiency. This model has a mean accuracy of 90% and a mean processing time of 8 seconds, which can replace human operators in the process. Additionally, the overall cost of the application is well under the general market price for this kind of solution.

In Ref. [11], the difficulties in handling backgrounds with noise, changes in lighting conditions, or complex textures in traditional image processing techniques useful for solving certain types of problems are highlighted. To solve this, this paper suggests that deep learning is widely used in automating defect detection and presents three methods to classify the efforts of the various literature on surface defect detection using deep learning technology. They are each based on the fault detection context, learning techniques, and fault localization and classification methods.

In Ref. [12], the existing object recognition methods through computer vision and deep learning did not perform well in detecting small and dense objects and failed even in images that were subjected to arbitrary geometric transformation. By comparing and analyzing algorithms, they propose an object detection network to solve the problem. The proposed method uses a deep convolutional network to obtain multi-scale feature values and adds a deformable convolutional structure to overcome geometric transformations. When only the existing YOLO was used, 48.47% of the MAP was derived, but when the proposed network was used, the result was improved to 52.55%.

In Ref. [13], as object detection technology using an artificial intelligence-based approach expands, the research team tries to use it to recognize objects in images and videos to respond to security issues. The team analyzed the robustness and integrity of object detection for future applications and state-of-the-art security research.

In Ref. [14], the problems caused by noisy data, a lack of labeled data, and computational complexity when using deep learning methods in industrial applications were analyzed. To solve these problems, the proposed crane casting recognition consisting of a lightweight dehazing method for pre-processing noisy videos to remove haze, fog, and smoke, and end-to-end corner casting recognition by applying a recurrent neural network along with long short-term memory (LSTM) units. The case study presents real-time container corner casting recognition for the efficient loading and unloading of the container. The proposed method is verified in the field with an average accuracy rate of 96%.

When analyzing the above studies, the defect detection methods using YOLO in the existing intelligent manufacturing research had the disadvantage that mAP showed a value that was not suitable for application to the actual process and was limited only to the product's surface. We propose a model that is not limited to the surface of an object, obtains an image of the contents of a transparent container from the outside through a camera, and checks the appropriate amount of the contents through object recognition from the obtained image. In this study, deep learning, computer vision, and OpenCV were applied to implement a defect detection model in the manufacturing process that SMEs can use without significant capital and technology.

2.2. YOLOv4

In this paper, the YOLOv4-tiny model [15,16] was used for the surface recognition of the liquefied gas charged in the lighter. Unlike R-CNN, YOLO can recognize objects faster because learning is performed with one CNN without a region proposal process. The basic idea of YOLO is to divide the input image into a grid of S*S that means product of S and S, and each grid cell predicts the bounding boxes and reliability of those bounding boxes. The reliability indicates the probability that a specific object appears in the bounding box and how well the predicted bounding box fits the object. Based on the predicted information, the bounding box is adjusted and classified. Finally, the object is recognized in a form that allows the object to exist entirely inside the bounding box. Detecting a defect with a non-uniform shape is difficult because the defect itself is not uniform and has a complex form, and each defect has its own unique characteristics. These defects may appear differently for each production line, and for this purpose, YOLO is used to detect defects.

3. Proposed Deep Learning Based on Defect Detection Method

In this paper, the target of error detection judgment is defined in two ways. The first is to determine the volume of contents in the transparent container. The second is to judge the error depending on the location and damage of the label sticker attached to the outside of the container. The following figure, Figure 1, is a model proposed for the above two kinds of defect.

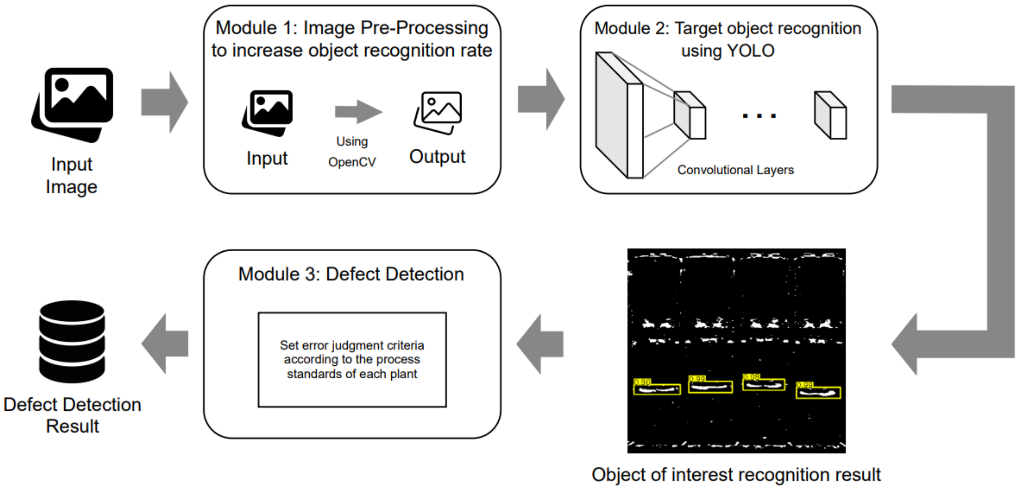

Input
Image
Module 1: Image Pre-Processing
to increase object recognition rate
Input
Using
OpenCV
Output
Module 2: Target object recognition
using YOLO
Convolutional Layers
Object of interest recognition result
Module 3: Defect Detection
Set error judgment criteria
according to the process
standards of each plant
Defect Detection
Result

Figure 1. A proposed deep learning-based defect detection model.

The captured input image is (1) pre-processed using OpenCV [17,18], and (2) pre-learned YOLO is applied to recognize the target object of the image. After extracting the ROI, the region of interest of the object, using the information on the extracted region, (3) an error is detected according to the error determination standard that meets the process standard of each factory set in advance. Through this model, it is possible to minimize the economic and technical burden of a small factory by allowing the model to be used to find any defects that can detect only with images taken from the outside without a scale or additional equipment.

In order to use the proposed model to detect defects, the following conditions must be established. First, the camera must be fixed so that the distance between the camera and the object is constant. Second, optimized lighting conditions for image shooting should enable work without affecting the surrounding lighting environment. Third, the camera's FOV should be set as close to that of a human as possible for accurate judgment. Finally, the distortion caused by the lens is removed using the distortion removal algorithm.

Depending on the judgment target, the following considerations are in the image processing and object recognition process. Intentional vibration is applied to simulate the moving on conveyor belts in factories, especially if the contents are fluid. The temporal and physical gaps that may occur due to the mechanical movement should be considered when the judgment target moves.

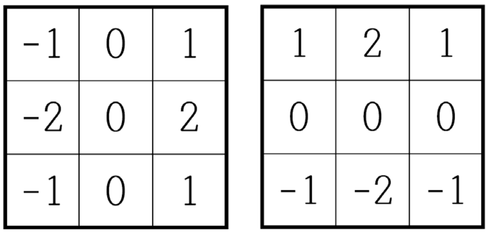
3.1. Module 1: Image Preprocessing

A preprocessing process is required to apply the defect determination algorithm using the model. First, image data to be judged were obtained, and only the region to be considered was extracted from the image. Since the object of judgment of the model proposed in this paper is the content in the container, and the edge of the content surface to measure the amount of content in the container, the outline was sharpened through the image processing process using OpenCV to facilitate recognition. The RGB-scale image was converted into a black-and-white image to apply the Sobel edge detection algorithm. If the brightness of each RGB scale in an arbitrary pixel was R, G, B, the grayscale brightness value in the corresponding pixel was calculated by using Equation (1).

$$\text{gray} = \frac{\mathrm{R} + \mathrm{G} + \mathrm{B}}{3} \tag{1}$$

The edge of the surface of the contents inside the container was extracted by applying the Sobel edge detection algorithm to the converted black and white image. This algorithm is a process of catching the border line where the brightness is different in the picture, and through this process, the boundary of the part with the difference in brightness can be more clearly defined.

The Sobel edge detection algorithm [19] extracted the edges within the image by applying the masks shown in Figure 2 to the image. The vertical outline could be extracted through the vertical mask, as shown in Figure 2a, and the horizontal outline could be extracted through the horizontal mask, as shown in Figure 2b.



-1	0	1
-2	0	2
-1	0	1

(**a**)

1	2	1
0	0	0
-1	-2	-1

(**b**)

Figure 2. Default Sobel masks: (**a**) 3 × 3 vertical mask; (**b**) 3 × 3 horizontal mask.

The mask processing was performed as follows. When a certain pixel P(x, y) overlaps with the center of the mask, the absolute value of the added value by multiplying the brightness value of the pixel overlapping the mask by the values of the mask was set to a new brightness value P(x, y).

Through this process, the surface outline was extracted with a value brighter than the surroundings, so the periphery, except for the shape of the surface, was darkened using the binarization from OTSU's method [20] to facilitate recognition.

When the threshold value was set, the brightness value of pixels below the threshold value was changed to 0, and the brightness value of pixels above the threshold value was changed to 1.

OTSU's method, which is one of the methods of binarization, is a method of selecting an appropriate threshold value. When classifying the entire brightness area into an area with a brightness below the threshold value and another area having a brightness above the threshold value, a threshold value at which the variance between the two areas is maximized is found, or the variance within each area is minimized.

The weights and variances of the two areas are ω_0, σ_0^2 and ω_1, σ_1^2. When the threshold is k and the overall brightness level is L, σ_0^2 and σ_1^2 can be obtained as in Equations (2) and (3).

$$\sigma_0^2 = \sum_{i=1}^{k} [i - \mu_0]^2 P(i)/\omega_0 \tag{2}$$

$$\sigma_1^2 = \sum_{i=k+1}^{L} [i - \mu_1]^2 P(i)/\omega_1 \tag{3}$$

$P(i)$ denotes the ratio of pixels having an arbitrary brightness value i among all pixels. μ_0 and μ_1 denote each average of the two areas and can be obtained as in Equations (4) and (5).

$$\mu_0 = \sum_{i=1}^{k} iP(i)/\omega_0 \tag{4}$$

$$\mu_1 = \sum_{i=k+1}^{L} iP(i)/\omega_1 \tag{5}$$

The value added by multiplying the variance of the two areas by weight is called the within-class variance, and when expressed as σ_w^2, σ_w^2 can be obtained as in Equation (6).

$$\sigma_w^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2 \tag{6}$$

OTSU's method specifies the value k at which the variance within the class is the minimum as the threshold value. When judging defects, the image was used through the same process as above and used for learning. In this model, to use YOLO, which recognizes the object to be judged, the obtained image data were pre-processed through the above process and utilized.

3.2. Module 2: Object Recognition Using YOLO

The pre-processed image data used YOLO to recognize and label the surface of product contents to be judged to create a custom dataset. It was used to create a customized YOLO object recognition model for product defect recognition. At this time, a model was completed by learning the area of the object to be judged by reflecting various environments and constraints that may occur in the real factory environment. As the contents were floating liquid, a separate process was required.

3.3. Module 3: Defect Detection According to Product Characteristics

After each process described above, a defect determination algorithm specialized for a specific product was required to determine the final defect and output the determination result. To implement the algorithm, specific criteria for judging defects were required, and the criteria for judging defects that met the process standards set for each product were used. Finally, the final defect determination result was output, which was obtained from all the above-described processes.

If the model was applied by setting the image processing process and error determination criteria suitable for each factory's defect determination target, the suggested model could be used for the production level of common factory automation.

4. Defect Detection Method Applying the Proposed Model

4.1. Determination of Defect The Liquefied Gas Volume of Disposable Gas Lighters

We aimed to verify the accuracy of the proposed model and its validity for application to the manufacturing process of SMEs by using the previously proposed error detection model to the 'defect detection of the liquefied gas volume of disposable gas lighters'. Since this paper is the result of a project to be applied to actual industrial sites, the lighter was selected as the error detection model. In addition, if the model was applied by extending it to various products, the paper may have become extensive, and in the case of products other than lighters, consultation with the industrial site of the product is required, so this is left for future research. To find defective products, the conventional method of visually and manually identifying defective products has a limitation in that it cannot correctly search for defective products in about 200,000 lighters per day. For this reason, an unmanned automation system is required.

To apply to the actual process of judging the volume defect of liquefied gas of a disposable gas lighter, the proposed model needs to be modified to match the lighter manufacturing process. Figure 3 shows the modified model for the disposable gas lighter defect detection system.

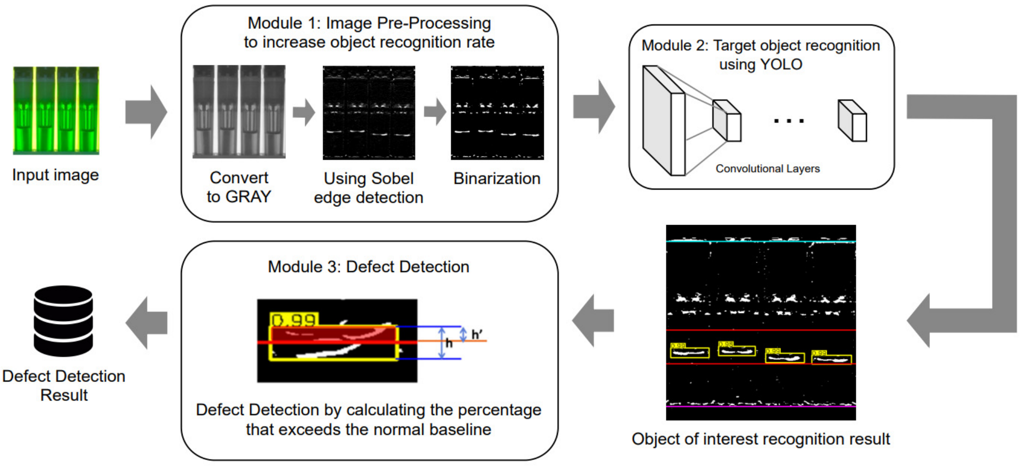

Module 1: Image Pre-Processing
to increase object recognition rate
Input image
Convert
to GRAY
Using Sobel
edge detection
Binarization
Module 2: Target object recognition
using YOLO
Convolutional Layers
Object of interest recognition result
Module 3: Defect Detection
0.99
h
h'
Defect Detection by calculating the percentage
that exceeds the normal baseline
Defect Detection
Result

Figure 3. The modified model applied to the gas lighter defect detection system.

4.1.1. Image Preprocessing

The proposed defective product detection system determines whether the product is defective by indirectly calculating the volume of transparent liquefied gas in the translucent fuel tank where the liquefied gas insertion process has been completed, based on the captured image.

To measure the volume of transparent liquefied gas inside a translucent plastic fuel tank with an image, it is necessary to take appropriate pre-measures. First, LED emitters were installed in front of and behind the gas lighter, which was the subject, so that the accuracy of the program was not adversely affected by the surrounding environmental factors, and the transparent liquefied gas was made to stand out. Figure 4 shows the image of the original lighter and the result of converting it through the image preprocessing process described above. Figure 4 shows only the four lighters in the center in the image of ten green lighters in one tray. In the actual process, an image with ten lighters was used.

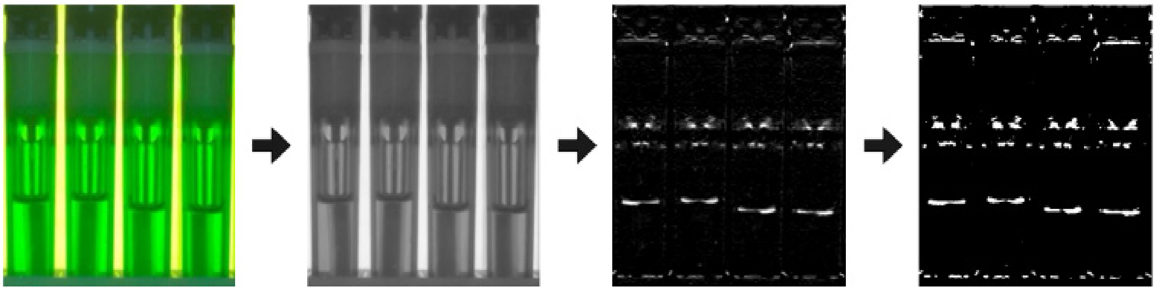


Figure 4. Result of liquefied gas image preprocessing.

4.1.2. Surface Object Recognition

After labeling the surface of liquefied gas using the surface image, as shown in Figure 4, machine learning was performed to create a YOLOv4-tiny object recognition model. The dataset for model learning was collected through images taken in the actual lighter manufacturing process, and a training dataset was created by labeling 1180 images of the surface of the liquefied gas inside the lighter. In the experiment, the batch size was trained to 64, the training result of the model showed precision = 0.96; recall = 0.89; F1-score = 0.92; average IoU = 76.71%; and mAP = 95.68, and the detection time was measured as 1 sec. In this paper, to use YOLOv4-tiny, a learning model for liquefied gas surface recognition inside the lighter was created using YOLOv4-DarkNet. YOLOv4-tiny is a compressed version of YOLOv4, which further simplifies the neural network structure and enables object recognition in mobile and embedded systems, so it is also suitable for application to Jetson-Nano, our experimental environment.

Through object recognition using the model, it is possible to quickly recognize the surface of transparent liquefied gas without being greatly affected by the surrounding environment, such as the illumination and colors of a lighter tank. For reference, the gas tank of the lighter used in the experiment was made in five colors: yellow, purple, red, green, and blue.

Figure 5 shows the results of the surface of the liquefied gas recognized through the previously described liquefied gas surface recognition model. The cyan line is the upper base line; the red lines are the upper normal reference line and lower normal reference line, respectively; the yellow boxes between the red lines are bounding boxes of the surface detection result; and the magenta line is the lower base line. It was determined whether the amount of gas in the lighter was defective according to whether the surface existed within the preset upper normal reference line and the lower normal reference line. To determine whether the surface was within the normal range, the insert of the lighter was set as the upper base line, and the upper end of the plastic tray containing ten lighters was designated as the lower base line.

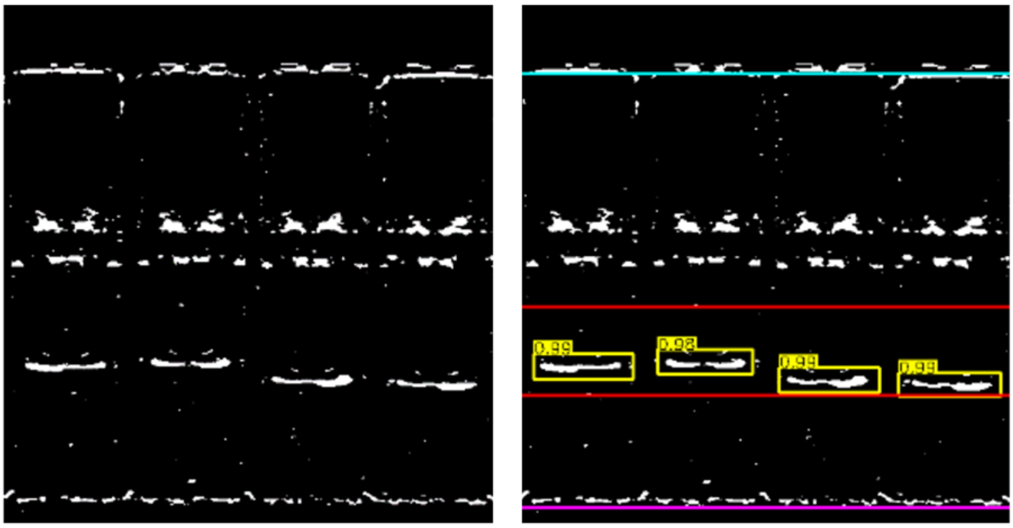


Figure 5. The surface of liquefied gas recognition result.

4.1.3. Determination of Defect

The volume of liquefied gas in the translucent fuel tank can be classified as over, under, and tiny. Tiny refers to a case in which the gas surface cannot be recognized because the amount of gas is too small for the camera to identify the surface of the liquefied gas. To correctly determine whether the various amounts of gas are defective, the validity of the indirectly measured volume and the established normal standard must be secured. Figure 6 shows the calculation of the normal area using the length of each part of the gas lighter measured using a measuring device.

According to the Korean Agency for Technology and Standards Notice No. 2010-530 "Safety standards for industrial products subject to safety certification", the volume of liquefied gas should not exceed 85% of the total volume, and if the total volume is less than 60%, it is judged as under. Assume that the rear surface of the liquefied gas is full, and if the front surface is 33 mm to 40 mm from the ground, the volume of liquefied gas occupies 60~85% of the total volume. The height of the tray is 22 mm and the viewable area of the gas surface is 26 mm, excluding the part where the gas is not visible because the insert and tray are opaque. Therefore, the lower reference line of the normal area is 11 mm above the normal base line, which is the upper part of the tray, so it is located at 11/26 of the height of the viewable area. The upper reference line of the normal area is 18 mm above the normal base line, so it is located at 18/26 of the height of the viewable area. Using this ratio, in an actual program, the normal area can be extracted from the pixel values between the two base lines. This is represented by a straight red line in Figure 5. If the liquefied gas surface bounding box recognized by the object recognition model is within the relevant area, it is judged as normal. If it exceeds the upper reference line, it is over, and if it does not exceed the lower reference line, it is judged as under. As a result, the total liquefied gas volume can be indirectly inferred through the height value of the front liquefied gas calculated from the image.

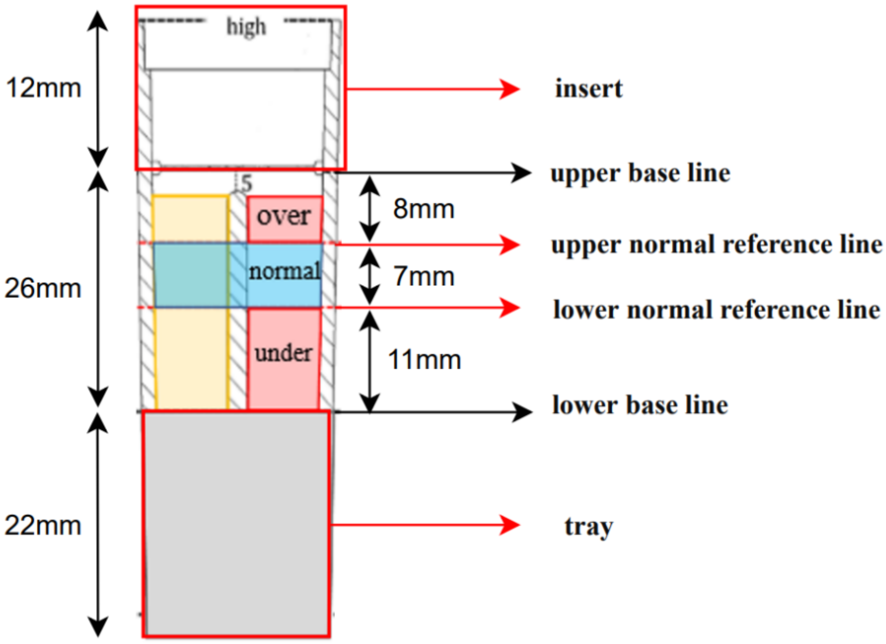
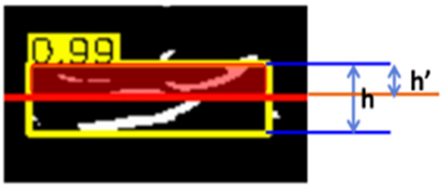


Figure 6. Normal area according to actual measurement.

Figure 7 shows the method of calculating the Rate of Out of Normal Range to objectively determine whether the detected liquefied gas surface is located at the upper or lower reference line. When the bounding box created by recognizing the surface exceeds the upper or lower reference line, the ratio of the exceeded height h′ to the box height h is called Rate of Out of Normal Range, and in this paper, it was called RONR. Figure 8 shows the flow chart of the judging algorithm using the calculated RONR. In this system, the *n*-th lighter is judged k times, and if the average value of the k RONR exceeds the normal threshold, the *n*-th lighter is judged as an over- or under-defective product. The size of the bounding box of the surface recognition is recognized as thicker than the actual surface for reasons, such as shaking or surface tension. Therefore, in this paper, the normal threshold value was set to 90% to detect only the defect.



$$\left(\sum_{i=1}^{k} \frac{h_i{}'}{h_i}\right)/k$$

Figure 7. Calculation method of RONR for judging defective products.

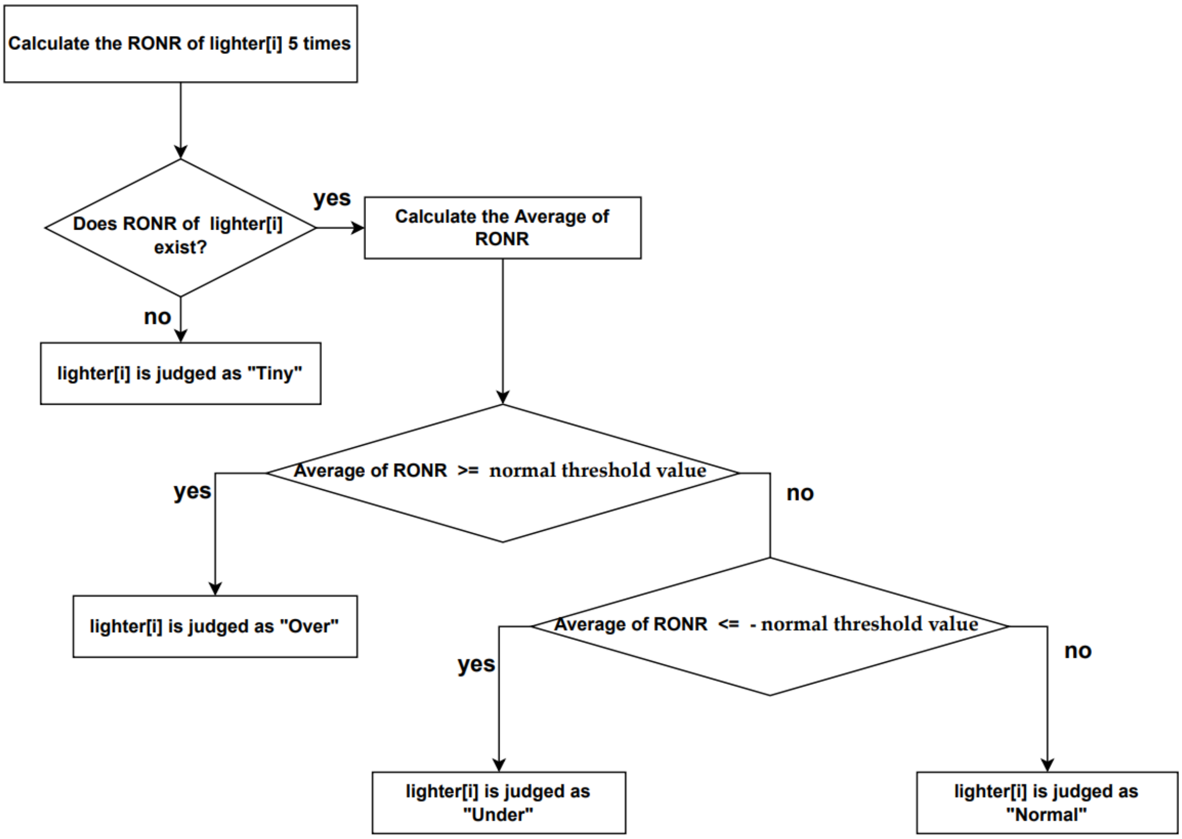

Calculate the RONR of lighter[i] 5 times
Does RONR of lighter[i] exist?
yes
Calculate the Average of RONR
no
lighter[i] is judged as "Tiny"
Average of RONR >= normal threshold value
yes
no
lighter[i] is judged as "Over"
Average of RONR <= - normal threshold value
yes
no
lighter[i] is judged as "Under"
lighter[i] is judged as "Normal"

Figure 8. Algorithm flow chart for judging defective products.

5. Defect Detection Experiment Applying the Proposed Model

5.1. Experimental Environment

Figure 9 shows the experimental environment in which the camera, front emitter, rear emitter, and polarization panel were installed at a distance applicable to the actual process. As shown in Figure 9, the front emitter was installed to illuminate the front part of the lighter set at 50 mm from the ground, and the rear emitter was installed to illuminate the rear part of the lighter set at a location 50 mm away from the lighter set. A polarizing panel was used between the rear emitter and the lighter set so that the light was evenly distributed on the back of the lighter set. Experiments were conducted with the same illuminance for each color of the lighter. The distance between the camera and the lighter set was set to 330 mm, which is the same distance applied in the actual process. To test the shaking case, a mini conveyor belt experiment environment similar to the factory conveyor belt was made, and the shaking experiment of the liquefied gas inside the lighter was conducted. Figure 10 is an actual picture of the conveyor test environment.

Table 1 below shows the types of hardware and software versions used in the experiment.

Table 1. Hardware type and software version.

Hardware	Software
Nvidia Jetson NANO B01	Nvidia JetPack 4.5.0
Raspberry Pi HQ Camera 12.3 MP	OpenCV 4.5.2
16 mm Telephoto Lens for Raspberry Pi HQ Camera	Python 3.8.9
LED floodlight 50 W	YOLO v4

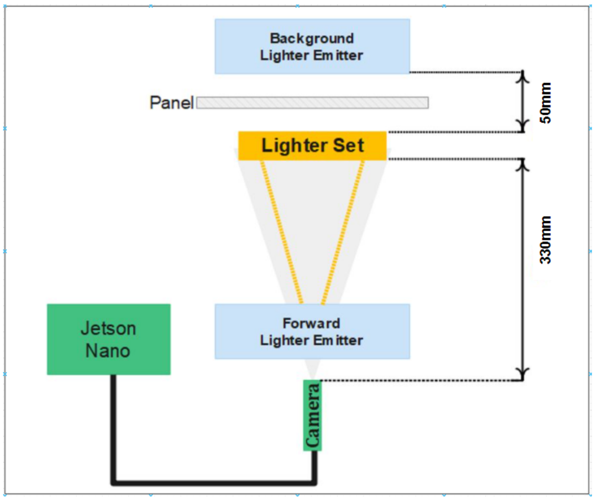


Figure 9. The experimental environment.

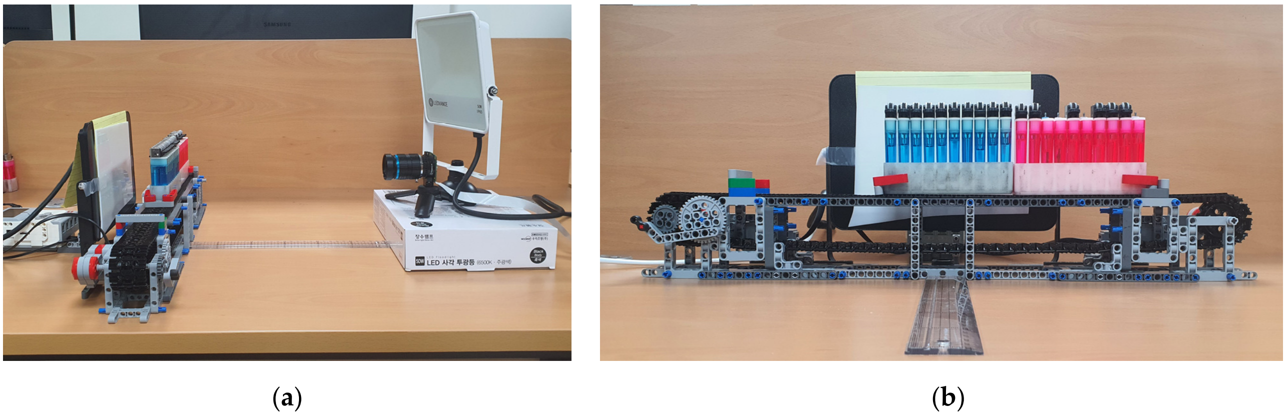


Figure 10. Conveyor experiment environment: (**a**) side of conveyor; (**b**) front of conveyor.

5.2. Volume of Gas Defect Judgment Experiment

This experiment aimed to verify the practical usage of the system proposed in this paper by measuring the accuracy of detecting a total of four cases ('normal'; 'over'; 'under'; and 'tiny', a kind of under, but the amount of content is too small or nonexistent, so the surface does not appear in the image; thus, tiny was treated separately from under.) that may occur in the actual process.

5.2.1. Experiment Method

For lighters with a total of 5 colors, yellow, purple, red, green, and blue, a total of 10 lighters within the normal range were configured as one test set; lighters from 1 to 10 were sequentially under, over, and tiny; and the remaining seven lighters were composed of lighters with normal cases, and according to each case, 1000 times, a total of 50,000 experiments were performed. A total of 50,000 experiments were performed for the conveyor experiment using the same test set as above.

5.2.2. Experiment Result

Table 2 shows the results of an experiment to find out the difference in recognition rate by changing the illuminance while increasing the distance between the lighter and the rear emitter that illuminates the lighter from the rear by 5 cm to examine the relationship between illuminance and object recognition. For this experiment, using the same test set as the two experiments above, 1500 experiments at each distance for a total of 6 distances from 5 cm to 30 cm, a total of 9000 experiments were performed. As shown in Table 2, the recognition rate of the liquefied gas inside the disposable gas lighter in the real-time video was the highest at 99.87% in the case of 10 cm, and the gas volume judgment accuracy was the highest at 99.33% in the case of 5 cm. However, since the recognition rate was not significantly different between the case of 5 cm and the case of 10 cm, the optimal illuminance condition was set to 5 cm in consideration of both accuracy and recognition rate. The reason that the recognition rate decreases as the distance increases is that the lower the illuminance, the more difficult it is to distinguish the surface of the liquid gas inside the lighter from the surface of the lighter. This is the main cause of lowering the accuracy of judgment.

Table 2. Experiment result 1.

State of Lighter	**Object Recognition Result**	**Distance between Lighter Set and Emitter**					
		5 cm	**10 cm**	**15 cm**	**20 cm**	**25 cm**	**30 cm**
Normal	Accuracy/% Recognition rate/%	99.62 99.81	98.57 99.81	86.57 86.95	46.48 46.76	40.95 41.14	22.57 22.67
Over	Accuracy/% Recognition rate/%	98 100	99.33 100	90 90	100 100	99.33 100	84.67 92.67
Under	Accuracy/% Recognition rate/%	98.67 99.33	99.33 100	95.33 95.33	96 96	96.67 97.33	86 90
Tiny	Accuracy/% Recognition rate/%	98.67 -	96.67 -	100 -	100 -	99.33 -	98 -
Total accuracy		99.33	98.13	89.2	62.2	57.13	34.07
Total recognition rate		99.8	99.87	89.4	62.33	58.53	44.13

Table 3 shows the results according to a total of 50,000 experiments. The blue cell in the table means that the state of the lighter and the judgment result are the same, and the red cell means that the state of the lighter and the judgment result are not the same. The yellow lighter demonstrated 100% accuracy in normal, over, and under, and there was one error of recognizing a tiny amount as exceeding. The purple lighter showed 100% accuracy in normal, over, and tiny, and there were three errors of identifying under as tiny. Red, green, and blue lighters all showed 100% detection accuracy. In a total of 50,000 experiments, four errors occurred, showing an accuracy of 99.992%. The case of the conveyor shaking test showed a similar pattern to the above test result and showed an accuracy of 99.986%. From the results of this experiment, we can conclude that the shaking of the conveyor does not significantly affect the recognition of the liquefied gas surface.

Table 4 is the confusion matrix prepared based on the experimental results. In the test, all cases where the amount of gas was normal were recognized as normal, all cases of defective gas were identified as defective, and both precision and recall were 100%. However, there were cases in which a case of a specific defect was recognized as a case of another defect separately. In the case of an error in the yellow lighter, an error occurred in that the median strip of the lighter was recognized as a gas surface, and the case of tiny was detected as an over case. It is judged that the case of under was caught as a tiny case.

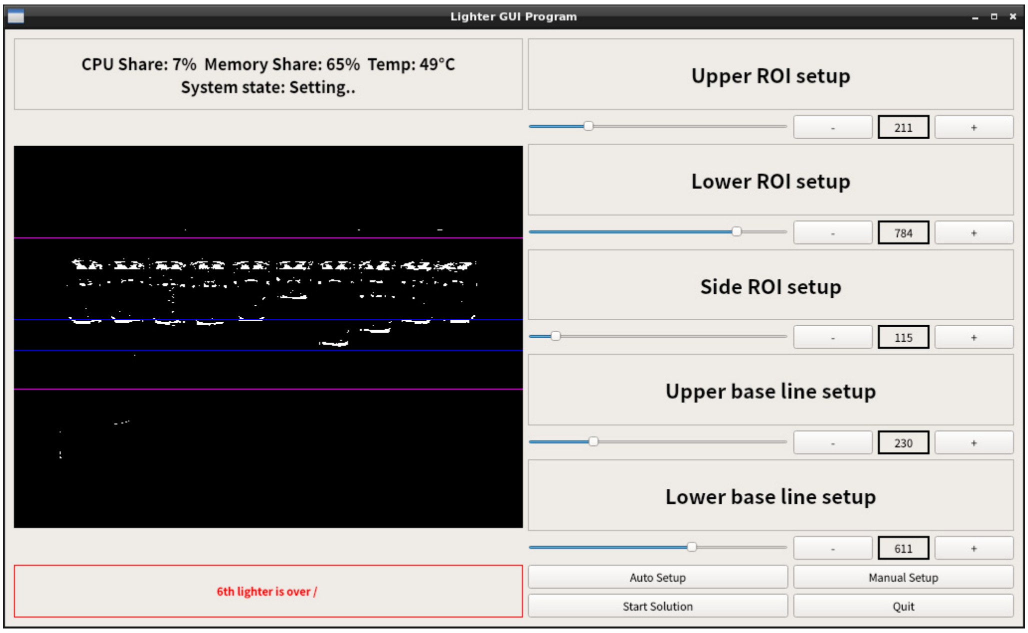
Since a set of lighters moves in front of the camera every 1.5 seconds to take an image, to detect defects using the system in an actual factory, defect detection must be completed within 1.5 seconds. In the 2500 additional processing time experiments conducted for processing speed analysis, the maximum value was 1.348 seconds, the minimum value was 0.886 seconds, the average was 0.904 seconds, and the standard deviation was 0.028 seconds, so it could be recognized within 1.5 seconds in all experiments. Additionally, in an actual program, if there is a defective lighter for a lighter set, the number of the defective lighter and the defective condition is output, as shown in Figure 11 below.

Table 3. Experiment result 2.

Lighter Color	Actual Gas Volume	Detection Result			
		Normal	Over	Under	Tiny
Yellow	Normal	7000	0	0	0
	Over	0	1000	0	0
	Under	0	0	1000	0
	Tiny	0	1	0	999
Purple	Normal	7000	0	0	0
	Over	0	1000	0	0
	Under	0	0	997	3
	Tiny	0	0	0	1000
Red	Normal	7000	0	0	0
	Over	0	1000	0	0
	Under	0	0	1000	0
	Tiny	0	0	0	1000
Green	Normal	7000	0	0	0
	Over	0	1000	0	0
	Under	0	0	1000	0
	Tiny	0	0	0	1000
Blue	Normal	7000	0	0	0
	Over	0	1000	0	0
	Under	0	0	1000	0
	Tiny	0	0	0	1000

Table 4. Confusion matrix for experimental results.

Classification Result	Real Result			
	Normal	Over	Under	Tiny
Normal	35,000	0	0	0
Over	0	5000	0	0
Under	0	0	4997	3
Tiny	0	1	0	4999



CPU Share: 7% Memory Share: 65% Temp: 49°C
System state: Setting..

Upper ROI setup 211

Lower ROI setup 784

Side ROI setup 115

Upper base line setup 230

Lower base line setup 611

6th lighter is over /

Figure 11. Output window of actual program.

6. Conclusion and Future Research

This study proposed an automatic defect detection model using the YOLOv4 deep learning object recognition model and OpenCV real-time image processing open source. This model is applicable for externally judging the appropriate amount of contents of the transparent container or to judging whether the attached product label is defective. In addition, using the proposed model, we developed a system that automatically detects defective products in the volume of liquefied gas injected in the manufacturing process. The model's accuracy was verified in the test case under environmental conditions as in the actual process. This model can reduce the manpower for defect detection, which can reduce labor costs, and can detect defects faster and more accurately than the existing detection method, so an increase in production over time can be expected. In the future, we plan to develop a lightweight hardware structure optimized for production so that it can be finally applied in a real factory environment and distributed as a sustainable computing system. In addition, data on error detection will be recorded to continuously improve the performance of the proposed model.

Author Contributions: Conceptualization—Y.-S.S.; data curation—S.-H.P. and K.-H.L.; formal analysis—S.-H.P. and K.-H.L.; investigation—S.-H.P. and K.-H.L.; methodology—all authors; project administration—Y.-S.S. and J.-S.P.; Software—S.-H.P. and K.-H.L.; supervision—Y.-S.S.; validation—S.-H.P. and K.-H.L.; visualization—S.-H.P. and K.-H.L.; original draft—S.-H.P. and K.-H.L.; review and editing—Y.-S.S. and J.-S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science, ICT), Korea, under the High-Potential Individuals Global Training Program (2021-0-01549) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, B.; Wan, J.; Shu, L.; Li, P.; Mukherjee, M.; Yin, B. Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges. *IEEE Access* **2017**, *6*, 6505–6519. [CrossRef]
2. Jeong, Y.S.; Park, J.H. Artificial Intelligence for the Fourth Industrial Revolution. *J. Inf. Process. Syst.* **2018**, *14*, 1301–1306.
3. Kuang, Z.; Tie, X. Computer Vision and Normalizing Flow Based Defect Detection. *arXiv* **2012**, arXiv:2012.06737.
4. Hatab, M.; Malekmohamadi, H.; Amira, A. Surface Defect Detection Using YOLO Network. In Proceedings of the SAI Intelligent Systems Conference, London, UK, 3–4 September 2020; Volume 1250, pp. 505–515.
5. Zheng, Z.; Zhao, J.; Li, Y. Research on Detecting Bearing-Cover Defects Based on Improved YOLOv3. *IEEE Access* **2021**, *9*, 10304–10315. [CrossRef]
6. Wang, T.; Yao, Y.; Chen, Y.; Zhang, M.; Tao, F.; Snoussi, H. Auto-Sorting System Toward Smart Factory Based on Deep Learning for Image Segmentation. *IEEE Access* **2018**, *18*, 8493–8501.
7. Lin, S.Y.; Li, H.Y. Integrated Circuit Board Object Detection and Image Augmentation Fusion Model Based on YOLO. *Front. Neurorobot.* **2021**, *15*, 762702. [CrossRef] [PubMed]
8. Hussain, N.; Khan, M.A.; Kadry, S.; Tariq, U.; Mostafa, R.R.; Choi, J.I.; Nam, Y.Y. Intelligent Deep Learning and Improved Whale Optimization Algorithm Based Framework for Object Recognition. *Hum.-Cent. Comput. Inf. Sci.* **2021**, *11*, 34.
9. Yang, J.; Li, S.; Wang, Z.; Yang, G. Real-time tiny part defect detection system in manufacturing using deep learning. *IEEE Access* **2019**, *7*, 89278–89291. [CrossRef]
10. Andrei, A.T.; Henrietta, D.E. Low Cost Defect Detection Using a Deep Convolutional Neural Network. In Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 21–23 May 2020; pp. 1–5.
11. Bhatt, P.M.; Malhan, R.K.; Rajendran, P.; Shah, B.C.; Thakar, S.; Yoon, Y.J.; Gupta, S.K. Image-based surface defect detection using deep learning: A review. *J. Comput. Inf. Sci. Eng.* **2021**, *21*, 040801. [CrossRef]
12. Cao, D.; Chen, Z.; Gao, L. An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks. *Hum.-Cent. Comput. Inf. Sci.* **2020**, *10*, 1–22. [CrossRef]
13. Kim, K.H.; Jung, I.Y. Secure Object Detection Based on Deep Learning. *J. Inf. Process. Syst.* **2021**, *17*, 571–585.

14. Maqsood, M.; Mehmood, I.; Kharel, R.; Muhammad, K.; Lee, J.C.; Alnumay, W.S. Exploring the Role of Deep Learning in Industrial Applications: A Case Study on Coastal Crane Casting Recognition. *Hum.-Cent. Comput. Inf. Sci.* **2021**, *11*, 1–14.
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
16. Bochkovskiy, A.; Wang, C.Y.; Liao, M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
17. Jo, S.J.; Kwon, S.H.; Hwang, S.J.; Hwang, H.Y.; Yoo, J.Y.; Shin, S.Y. Mean area detection in the image using OpenCV. In Proceedings of the Korean Institute of Information and Communication Sciences Conference, KIICE, Jeonju, Korea, 31 May 2018; pp. 182–183.
18. Jang, S.H.; Jeong, J.P. Design and Implementation of OpenCV-based Inventory Management System to build Small and Medium Enterprise Smart Factory. *J. Inst. Internet Broadcast. Commun.* **2019**, *19*, 161–170.
19. Sobel, I. An isotropic 3 × 3 image gradient operator. In *Machine Vision for Three-Dimensional Scenes*; Freeman, H., Ed.; Academic Press: New York, NY, USA, 1990; pp. 376–379.
20. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]