An implementation of YOLO-family algorithms in classifying the product quality for the acrylonitrile butadiene styrene metallization

Yuh Wen Chen¹ · Jing Mau Shiu²

Received: 10 September 2021 / Accepted: 4 January 2022 / Published online: 29 January 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

In the traditional electroplating industry of Acrylonitrile Butadiene Styrene (ABS), quality control inspection of the product surface is usually performed with the naked eye. However, these defects on the surface of electroplated products are minor and easily ignored under reflective conditions. If the number of defectiveness and samples is too large, manual inspection will be challenging and time-consuming. We innovatively applied additive manufacturing (AM) to design and assemble an automatic optical inspection (AOI) system with the latest progress of artificial intelligence. The system can identify defects on the reflective surface of the plated product. Based on the deep learning framework from You Only Look Once (YOLO), we successfully started the neural network model on graphics processing unit (GPU) using the family of YOLO algorithms: from v2 to v5. Finally, our efforts showed an accuracy rate over an average of 70 percentage for detecting real-time video data in production lines. We also compare the classification performance among various YOLO algorithms. Our visual inspection efforts significantly reduce the labor cost of visual inspection in the electroplating industry and show its vision in smart manufacturing.

Keywords Deep learning · Acrylonitrile Butadiene Styrene (ABS) · You Only Look Once (YOLO) · Automatic Optical Inspection (AOI) · Additive manufacturing · Artificial intelligence · Smart manufacturing

1 Introduction

The rapid development of smart manufacturing, assisted by advanced information and manufacturing technologies [1], enables and speeds all physical processes and cyber flows to be available when and where they are needed across supply chains. A global integration for innovation for multiple small and medium-sized enterprises (SMEs) and large companies [2, 3] is critical to achieving this goal, particularly

the innovation of the machinery industry based on artificial intelligence (AI) and the internet of things (IoT).

The industry of water hardware is also eager for digital transformation by smart manufacturing, and the metallization of plastics such as Acrylonitrile Butadiene Styrene (ABS) plastic is popularly and widely used. ABS has the best electroplating effect and is the most commonly used material because the metal ions plated are not easy to fall off [4]. However, plastic is usually a poor conductor and must first be attached to its conductive membrane. At present, the processing technology uses chemical plating or electroless plating to adhere a metal film to the surface of the plastic, followed by subsequent copper plating or nickel plating (and chrome plating). Therefore, we should seriously monitor the final quality of the ABS product under the control of the process.

Automatic optical inspection (AOI) [5] is very common in quality inspection applications. Generally speaking, production lines need to use AOI for inspection, mainly for products with large and stable production volumes or products with rigorous product quality inspection. Representatives

✉ Yuh Wen Chen
profchen@mail.dyu.edu.tw

Jing Mau Shiu
a979319516@gmail.com

¹ Department of Information Management, Da-Yeh University, No.168, University Rd., Dacun, Changhua 515006, Taiwan, R.O.C.

² Department of Industrial Engineering and Management, Da-Yeh University, No.168, University Rd., Dacun, Changhua 515006, Taiwan, R.O.C.

of such industries include printed circuit boards (PCBs), semiconductors, cell phone parts, medical devices, etc. However, with the increasing number of defect detection items, traditional AOI requires the continuous development of new algorithms for different defect characteristics, which is costly and takes a relatively long time to develop. In addition, the algorithm of the traditional program is susceptible to noise interference, and it is less stable against external interference. However, detecting reflective objects, such as electroplated objects, is hard to judge the defects due to the reflection of a metal surface. In contrast, deep learning in artificial intelligence is trained by thousands or even millions of defect photos, which is more flexible and resistant to external interference than traditional AOI. Therefore, we study this interesting issue based on the YOLO (You Only Look Once) framework of electroplated object detection [6–8]. YOLO is a state-of-the-art, real-time object detection system [9], YOLO is developed on a single convolutional network, which simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance [10]. This unique detection system has revealed its power in many fields; for example, Li et al. [11] used YOLO v3 to detect electronic components on PCB to achieve quality control of more than 90 percent accuracy. Tian et al. (2019) used YOLO to see fruits in orchards to judge apples' growth stages. Liu and Wang [12] proposed the YOLO to classify the diseases and pests for tomatoes. In addition, this YOLO is also valuable in mask detection during COVID-19 [13].

Our methodology has the following steps:

1. We designed the AOI prototype for the customized needs of the study. We developed an AOI prototype and assembled it with additive manufacturing (AM) technology. The AOI prototype is created by ourselves to detect the defects on electroplated ABS products, integrated with Arduino C++ programming to control the motor, inspection light source, stepping motor, robot arm, other related electronic parts, etc.
2. We use the AOT prototype to capture the necessary videos and divide them into frames per second (FPS) to label the defective images. The images are classified into two categories: defective and non-defective. We apply the Python Open Source Computer Vision Library (OpenCV) for image pre-processing, and the processed images are labeled with defect types and locations on each product image.
3. We set up the deep learning environment on the graphics processing unit (GPU) to apply different YOLO algorithms to compare their performance for the same validation set of images. The video captured from the AOI

prototype is sent to YOLO models for deep learning to classify products' defects in real-time. After that, we train the customized YOLO weights by our dataset to classify the defects of products among various YOLO algorithms: from v2 to v5 in this study.

4. Finally, the indexes of receiver operating characteristic (ROC) compare the detecting performance from different YOLO frameworks.

We organize this study as follows. Section 1 briefly describes the background and motivation of the study; Section 2 explains the hardware and software tools and theoretical methods used. Section 3 describes the process of this study, the description of the self-assembled machine and the calculation results, and Sect. 4 provides the conclusion and recommendations. In brief, our contributions are summarized as follows:

1. We assemble a customized AOI assisted by three-dimensional (3D) additive manufacturing (AM) to detect the defectives of electroplated products (ABS).
2. The algorithm of AOI is launched by the deep learning schema from the YOLO family: from v2 to v5.
3. We compare the performance of various versions of YOLO and discuss their characteristics.

This study aims to develop an advanced AOI paradigm by AI for smart manufacturing nowadays. Our efforts integrate AI and AOI as advanced manufacturing technology and inspire the electroplating industry in tradition.

2 Methodology

We try to find the papers integrating YOLO algorithms and AOI for the qualitative detection of the electroplating industry; however, the paper number of this field is relatively less. Most of these contributions focus on the initiative of PCB. Therefore, this study should be valuable for the smart manufacturing of small and medium enterprises (SMEs). The methodology of this study is illustrated in Fig. 1. First, we designed the AOI prototype for the customized needs by CAD. At the same time, we review the necessary literature for AOI and YOLO. Second, we assemble the AOI prototype by 3D additive manufacturing, and various versions of YOLO versions are developed in PyCharm (<https://www.jetbrains.com/pycharm/>). Third, the AI-based AOI platform integrates the trained YOLO models. Fourth, the AOT prototype captures the necessary images. The images are classified into two categories: defective and

Fig. 1 Research Process

non-defective. Fifth, the GPU tests different YOLO algorithms to compare their performance for the same validation set of images. Various YOLO algorithms: from v2 to v5, are tested in this study. Finally, the indexes of receiver operating characteristic (ROC) compare the detecting performance from different YOLO frameworks.

2.1 Automated optical inspection

Automated optical inspection (AOI) is a visual inspection where a camera autonomously scans the failures of products, shown in Fig. 2. AOI system early application is in textile, packaging, automotive, machinery, etc. The main focus is

Fig. 2 Operation of AOI

on the speed of detection requirements, introducing the AOI inspection system, and significantly improving detection speed [14]. Still, this stage of the AOI inspection system detects larger objects and defects and must compete with the human eye. Hence, the development of the AOI inspection system stage simulates the naked eye performance and speed, limited by time. Although the pace of inspection can meet the requirements, there is no demand for too fine or high precision in the past AOI inspection items, and the development of the resolution of the inspection has not been greatly improved [15]. However, with the industrial development in Taiwan, technology industries such as semiconductors, electronics, biotechnology, and optoelectronics have emerged in the past two decades. Unlike traditional products, technology products are usually more complex, high-precision, tiny, and less tolerant, and any problems in the process may lead to errors in the whole system. In the technology products, the small size and lightweight characteristics, if the AOI optical image inspection, the required resolution is relatively precise to push the need of AOI [16]. The development of AOI has dramatically reduced the manual visual inspection error rate, thus improving production control and quality inspection speed, which is what the manufacturing industry expects. The improvement in resolution is also due to the introduction and application of the charge-coupled device (CCD) sensor. The image area and pixel resolution to get better data. However, considering CCD in price competition, the AOI system has the advantage of low cost. Therefore, many industries began to introduce AOI systems to replace manual inspection. In most of the technology industry and the demand for

inspection of production, the overall booming development of automated optical inspection, creating the growth and demand for the AOI industry. We can find some research for AOI; for example, since AOI is very sensitive to the environment, Chen and Perng [17] proposed a system of coaxial light, a backlight, and motion control to capture the characteristics of statistical textures of the molding surface.

Acciani et al. [18] developed the diagnosis as a pattern recognition problem with a neural network (NN) approach. Five types of solder joints have been classified to diagnose with a high recognition rate by NN. Richter and Streitferdt [19] proposed a smart AOI paradigm. This paradigm uses active and unsupervised learning superposition to build a fully annotated dataset. In addition, they train a suitable classifier by a deep learning cluster analysis. As the development of hardware is fast and deep learning technology is more popular today, the AOI combined with an evolutionary algorithm such as NN will play an essential role in the smart sensing network [20].

2.2 YOLO Framework

In image processing, a kernel, or a so-called convolution matrix, is a small matrix used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between a kernel and an image [21, 22]. Consider an image has a matrix form given a width w and a height h to store the pixel value in (x, y) . Assume $z_1, z_2 \in (x, y)$, then the kernel is defined as a function k that for all $z \in R^2$ satisfies

Fig. 3 CNN Framework

$$k(Z_1, Z_2) = \langle \phi_1, \phi_2 \rangle^p \quad (1)$$

Here k is a matrix, p is a positive integer and $\langle \phi_1, \phi_2 \rangle^p$ is a power form between two vectors: $\phi(z_1)$ and $\phi(z_2)$. For example, if we take $p=2$, then we have

$$\langle \phi_1, \phi_2 \rangle^2 = \phi(z_2)^T [\phi(z_1) \phi(z_1)^T] \phi(z_2) \quad (2)$$

$\phi(z_i)$ satisfies the following condition:

$$\phi : z_i \longrightarrow F \quad (3)$$

where z is the input space and F is the output/feature space. For example, taking the neural network model, the input nodes are z , and the final weighed outputs/nodes are F . For the two-dimensional data, let S be a given finite set and

$$S = [z_1, z_2, \dots, z_l] \quad (4)$$

we can propose the k in a matrix form as follows:

$$K^p = \begin{bmatrix} \langle \phi(z_1), \phi(z_1) \rangle^p, & \langle \phi(z_1), \phi(z_2) \rangle^p, & \dots, & \langle \phi(z_1), \phi(z_l) \rangle^p \\ \langle \phi(z_2), \phi(z_1) \rangle^p, & \langle \phi(z_2), \phi(z_2) \rangle^p, & \dots, & \langle \phi(z_2), \phi(z_l) \rangle^p \\ \vdots & \vdots & \ddots & \vdots \\ \langle \phi(z_l), \phi(z_1) \rangle^p, & \langle \phi(z_l), \phi(z_2) \rangle^p, & \dots, & \langle \phi(z_l), \phi(z_l) \rangle^p \end{bmatrix} \quad (5)$$

Here Convolution K^p is a general purpose filter effect for images. Simply speaking, we can extract the information from image by setting the filter and stride of kernel, and the padding of feature. Therefore, a basic framework of convolutional neural network (CNN) is shown as follows [23]:

This CNN structure was first proposed by Fukushima [24]. The development of CNNs is to solve the formidable challenge of image recognition and detection. A CNN consists of an input layer: we may recall the input vector z

above: a vector of pixels to represent an image; moreover, one or more hidden layers, and an output layer (Fig. 3). At the beginning of CNN, semantic segmentation is necessary. Segmentation network classifies every pixel in an image, resulting in an image that is segmented by class. YOLO is an abbreviation of “You Only Look Once.” YOLO algorithm uses neural networks to provide real-time object detection. The critical components of YOLO algorithms are CNNs [25]. The basic framework of YOLO is shown in Fig. 4 by its input image presented by pixels, processing segmentation (also by pixels) of the input image, and the layers of CNN for each stage [9].

YOLO can determine the class and location of the object in the image with only one CNN, which significantly improves the recognition speed. The whole network design is point-to-point, easy to train, and fast. The critical difference between CNN [26] and traditional neural networks is that the human retina inspires CNNs’ visual system. CNN models are validated to perform excellently in image processing, such as LeNet [27], the progenitor of CNNs, and AlexNet [28], which started to make a name for itself in the field of machine vision. In 1998, Yann LeCun et al. published “LeNet” as the original CNN framework, which formally brought the concept of convolution into machine learning. The LeNet network architecture uses MaxPooling to strengthen the features, and the output layer uses Radial Basis Function radially oriented Euclidean distance function. The distance function is used in the output layer.

YOLO requires a neural network framework for training, and for this, we have used DarkNet (<https://pjreddie.com/darknet/yolo/>). The first version of the input image of 448x448 has 26 layers, with 24 Convolution Layers

Fig. 4 YOLO Framework

followed by two fully connected layers. The major problem with The first YOLO: YOLOv1 is limited to its inability to detect microscopic objects. After the first version, many researchers and scholars extend the source code for better performance. After the first version, many researchers and scholars develop the source code for better performance. For example, YOLOv2 includes batch normalization layers after each convolutional layer and has thirty layers compared to YOLOv1 of twenty-six layers. Moreover, this version introduced the concept of anchor boxes. Anchor boxes are predefined boxes provided by the user to DarkNet, which gives the network an idea about the relative position and dimensions of the objects to be detected. It has to be calculated using the training set objects. YOLOv3 evolves to have one hundred and six layers of neural network. And it can catch on three scales for detecting small to large size objects. This version provides nine anchor boxes: three per scale. Most important, the multiclass problem turned into a multilabel problem. And YOLOv3 is powerful in detecting tiny objects. YOLOv4 was recently released in 2020, and two Taiwanese scholars from Academia Sinica joined this project [29]. According to Bochkovskiy et al. [29] YOLOv4 is optimal for real-time object detection tasks because its performance lies on the Pareto optimality curve of the accuracy and speed of frames per second (FPS). YOLOv5 is funded by the open-source by Ultralytics (<https://ultralytics.com/>) [30]. Ultralytics was founded in 2014, spearheading several U.S. Intelligence Community (IC) and Department of Defense (DoD) initiatives in particle physics, data science, and artificial intelligence.

2.3 Prototype construction

We designed the AOI prototype by Computer Aided Design (CAD) and assembled it with parts from additive manufacturing technology (3D printing) for the customized joints, which is shown in Fig. 5 (from left to right: CAD file, prototype and conveyor).

We use the series of Arduino boards to connect the stepping motor, light-emitting diode (LED), Dobot robot arm (<https://www.dobot.cc/>), conveyor, and an ultrasonic sensor to finish the AOT prototype. We designed the prototype by CAD and assembled it with parts from additive manufacturing technology (3D printing) for the customized joints. We found the image quality is susceptible to our experimental environment. In this study, we try to capture the curved shape of the detected object; the defect characteristics of the curved shape will be challenged if a closed-circuit television camera (CCTV) lens is selected. Therefore, this study uses a telecentric lens [31] with parallel LED lights to set up the AOI device to capture the defect images by flattening this non-planar feature. Telecentric lens is designed in response to image size variation or image distortion and deformation caused by the different working distances of CCTV lenses. The Telecentric lens and CCTV lenses comparison are shown in Fig. 6(Left: CCTV, Right: Telecentric).

The characteristic of a telecentric lens is that it can obtain the same size image for the same size object, which we expect to measure at different working distances. And the telecentric lens is generally used for dimensional measurement or surface inspection on curved products. It is designed in response to image size variation or image distortion and deformation caused by the different working distances of CCTV lenses. The characteristic of a telecentric lens is that it can obtain the same size image for the same size object to be measured at different working distances and is generally used for dimensional measurement or surface inspection on curved and curved products.

Therefore, after several try-and-error experiments, we use a pair of bar LED with the trigger of the arrival of product waiting for detection. The light source plays a vital role in the inspection process in machine vision. The critical point of the lighting design is to highlight the essential features for detection, suppress the unimportant background. To obtain an acceptable image, we must understand the object's surface characteristics to be tested through try and error

Fig. 5 CAD and AOI Prototype

Fig. 6 Comparison of Telecentric Lens (Right) and CCTV Lens(Left)

experiments to select the appropriate light source to facilitate the lens to capture the defective features of the tested object. If the brightness of the light source is too high or too low, the type of light source is not suitable, the wrong way of lighting will significantly affect the final image of the camera. This particular product image capture is shown in Fig. 7 by setting a pair of bar LEDs.

The tested products are fed into this AOI machine by the conveyor. The stepping motors are used to move up and down to complete the light source's illumination at different angles; therefore, this interesting AOI machine can reveal the defects of the global surface of the product. At the

same time, OpenCV combined with the camera is used to take images and recognition. We design this AOI machine that the conveyor will transmit the ABS product one by one, and the camera will record the video during operation. Once the video is captured, the robot arm will remove the undesirable product at the end of the conveyor. The unsatisfactory situation makes the removal decision of the ABS product. All YOLO algorithms from v2 to v4 (<https://github.com/AlexeyAB>) are developed and implemented on the Nvidia GPU server based on Linux of the AI Center at Da-Yeh University. And the YOLO v5 is provided by Ultralytics (<https://ultralytics.com>). The preparation of image processing is introduced as

Fig. 7 LED Mechanism for Capturing the Product Image

Fig. 8 Preprocessing of Images: Enhanced (Left) and Sharpened (Right) with Different Kernels

Table 1 Model Parameters

Model Parameters	YOLOv2	YOLOv3	YOLOv4	YOLOv5
Batch	64	64	64	64
Max Batches	5000	5000	5000	1000
Learning Rate	10^{-5}	10^{-4}	10^{-4}	10^{-4}

The parameters are set according to the model scale

follows: first, we need to label the defective products by reducing the image scale to increase the recognition efficiency. This study uses LabelImg (<https://github.com/tzutalin/labelImg>) as an image marker tool to form a dataset for the deep learning model of defect images. The LabelImg is an image marker tool with open-source code. This file type can record the defect coordinates marked on the image and provide a deep learning model for training. Second, we set Hue, Saturation, Value (HSV) mask to filter the objects other than the plated objects to reduce the image noise by using the ConvertScaleAbs function in OpenCV (<https://opencv.org/>) to enhance the contrast. The

contrast enhancement is done by using the ConvertScaleAbs function, as shown in Fig. 8 (left). Finally, the enhanced images are sharpened by using the Filter2D convolutional algorithm, which is the pre-training labeling process of images, as shown in Fig. 8 (right). This study collected 508 images including surface impurities from electroplated parts. The training, testing, and validation ratio of these images is 60 percent, 30 percent, and 10 percent, respectively. We labeled the data as two categories: the defective category, including speck and pock, and the non-defective category, to build the training, testing, and validating set. Once the data above is ready, we apply four YOLO algorithms: v2, v3, v4, and v5 to validate the discrimination power among various algorithms.

3 Experimental results

We collect 508 images for this study, and the input size of the image is 416 x 416. The model parameters of YOLO algorithms are summarized as in Table 1; the corresponding

Fig. 9 Loss Evolution with Iterations of YOLO v2

Fig. 10 Loss Evolution with Iterations of YOLO v3

parameters are customized to prevent breakdown/non-convergence during training. The iteration number is enlarged accordingly because YOLOv4 and YOLOv5 have more complicated frameworks than YOLOv3 and YOLOv2. Decision-makers should decide these parameters to meet their customized needs; this could be considered case by case.

Then we compare these YOLO models; for example, v2, v3, and v4. The loss values reduce steeper and faster during iterations from v2 to v4: see Figs. 9, 10 and 11, the vertical axis is loss and the horizontal axis is iteration. The pictures of Figs. 9, 10 and 11 mean the model evolves more effectively from v2 to v4. The average losses of v2, v3, and v4 are 5.79, 4.91, and 2.73, respectively. The training performance of average loss also shows that the v4 is superior to v3, and the v3 is superior to v2. We also checked the YOLO v5, which is controversial compared with YOLO v4 by its originality. However, according to our experiment in this study, In terms of loss, YOLO v5 is better, but the discrimination

power of the two models is almost the same; we will present the results later. The YOLO v5 is much smaller, which is very suitable for edge computing of AI users. The evolution of loss of YOLO v5 is shown in Fig. 12.

A ROC curve is an effective tool to compare the performance of the YOLO-classification models of this study. The confusion matrix is the basis of the ROC curve, and it is also the most basic, intuitive, and simple method to measure the accuracy of classification models [32]. The predictive performance of the categorical models is statistically presented in four categories by confusion matrix, which is shown in Fig. 13.

We continue to evaluate the model's performance by calculating *Accuracy*, *Precision*, *Recall*, and *F1 – Score* through four categories of statistics and use *Recall* as the primary reference for identifying the version among YOLO models. Here are the mathematical definitions for *Accuracy*, *Precision*, *Recall*, and *F1 – Score*:

Fig. 11 Loss Evolution with Iterations of YOLO v4

Fig. 12 Loss Evolution with Iterations of YOLO v5

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \quad (6)$$

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (9)$$

Moreover, the evaluation results of various models are summarized in Table 2. Interestingly, the YOLOv3 and YOLOv5 performed in a balanced scope (over 70 percent) in Accuracy, Precision, Recall, and F1-Score, simultaneously.

Fig. 13 Confusion Matrix

Predict \ Actual	Actually Defective (Positive)	Actually Non-defective (Negative)
Predicted Defective (Positive)	Predicted as defective Actually is defective (True Positive, TP)	Predicted as defective Actually is non-defective (False Positive, FP)
Predicted Non-defective (Negative)	Predicted as non-defective Actually is defective (False Negative, FN)	Predicted as non-defective Actually is non-defective (True Negative, TN)

Table 2 Model Evaluation

Model Performance	YOLOv2	YOLOv3	YOLOv4	YOLOv5
TP	0.18	0.71	0.71	0.75
FP	0.00	0.17	0.31	0.24
FN	0.82	0.29	0.29	0.25
TN	1.00	0.83	0.69	0.76
Accuracy	59	71	70	75
Precision	100	81	69	76
Recall	18	77	69	75
F1-Score	30	77	69	75

Here TP+FN = 1 and FP+TN = 1

And the accuracy of YOLOv2 is somewhat lower than we expected. However, we found overfitting and overemphasized phenomena for the detected results from YOLOv4 (see Fig. 14). When we compare these caught pictures from v3, v4, and v5 altogether, as in Fig. 14, we can easily find too many (fake) points of speck and pock generated by v4, and we think this causes the slightly lower performance of YOLOv4. YOLOv5 also has the similar problem. We guess this issue resulting from the YOLOv4 and YOLOv5 has larger and complicated CNNs; however, the data of this study is limited. Therefore, if we don't provide adequate data, they won't perform as well as expected. The overfitting effect is more severe for YOLOv4 than YOLOv5.

Fig. 14 Detected Results from Various YOLO Algorithms (Left: v3, Middle: v4, Right: v5)

4 Conclusions and recommendations

In this study, we successfully solve the quality detection problem of the electroplating industry by combining the YOLO technology and the customized prototype of the AOI inspection platform. After the actual validation, YOLOv3 and YOLOv5 perform in a balanced scope with an accuracy rate of over 70 percent. Although YOLOv2 serves poor, we found that YOLOv2 is more accurate in the large area of pock detection (large object). A decision-maker can arbitrarily assign weights for multiple criteria, for example, Accuracy, Precision, Recall, and F1-Score, to select the final model to launch in practice.

The study results validate that YOLOv3 and YOLOv5 have the advantage of overall other models combined with the AOI system for better quality detection. We assume YOLOv4 will only show its value and power if we can provide enough data. Shortly, setting the model parameters is a primary task worthy of exploration. In addition, more defective images should be automatically collected to reduce the overfitting/overemphasized phenomena for YOLOv4 and YOLOv5.

Finally, we recommend increasing the number of pieces for each defect model and controlling the difference in the number of samples for each defect to provide the model with valuable training data. We believe our efforts play an important role for SMEs toward smart manufacturing.

Author contributions We (two authors) contributed equally to this work.

Funding information This research is an independent research from my lab H560 in Da-Yeh University (no official funding).

Availability of data and materials Yes (after acceptance).

Code availability Yes (after acceptance).

Declarations

Consent for publication We agree with the Copyright Transfer Statement later.

Consent to participate We declared that we contribute equally to this work.

Ethics approval We declare this research is original and innovative.

Conflict of interest We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Jung K, Morris KC, Lyons KW, Leong S, Chao H (2015) Mapping strategic goals and operational performance metrics for smart manufacturing systems. *Proc Comp Sci* 44:184–193. <https://doi.org/10.1016/j.procs.2015.03.051>
2. Wan J, Tang S, Li D, Wang S, Liu C, Abbas H, Vasilako AV (2017) A manufacturing big data solution for active preventive maintenance. *IEEE Trans Industr Inf* 13:2039–2047. <https://doi.org/10.1109/TII.2017.2670505>
3. Wang SY, Wan J, Li D, Zhang C (2016) Implementing smart factory of Industrie 4.0: An outlook. *Int J Distrib Sens Netw* 12:1–16. <https://doi.org/10.1155/2016/3159805>
4. Teixeira LAC, Santini MC (2005) Surface conditioning of ABS for metallization without the use of chromium baths. *J Mater Process Technol* 170:37–41. <https://doi.org/10.1016/j.jmatprotec.2005.04.075>
5. Tsai YH, Lyu NY, Jung SY, Chang KH, Chang JY, Sun CT (2019) Deep Learning Based AOI System with Equivalent Convolutional Layers Transformed from Fully Connected Layers. In 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (pp. 103–107). IEEE. <https://doi.org/10.1109/AIM.2019.8868602>
6. Tian Y, Yang G, Wang Z, Wang H, Li E, Liang Z (2019) Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comp Electron Agric* 157(1):417–426. <https://doi.org/10.1016/j.compag.2019.01.012>
7. Xie Y, Cai J, Bhojwani R, Shekhar S, Knight J (2020) A locally-constrained yolo framework for detecting small and densely-distributed building footprints. *Int J Geogr Inf Sci* 34(4):777–801. <https://doi.org/10.1080/13658816.2019.1624761>
8. Liu E, Chen K, Xiang Z, Zhang J (2020) Conductive particle detection via deep learning for ACF bonding in TFT-LCD manufacturing. *J Int Manuf* 31(4):1037–1049. <https://doi.org/10.1007/s10845-019-01494-9>
9. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788). <https://doi.org/10.1109/CVPR.2016.91>
10. Plastiras G, Kyrkou C, Theocharides T (2018) Efficient convnet-based object detection for unmanned aerial vehicles by selective tile processing. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3243394.3243692>
11. Li J, Gu J, Huang Z, Wen J (2019) Application research of improved YOLO V3 algorithm in PCB electronic component detection. *Appl Sci (Switzerland)*, 9(18). <https://doi.org/10.3390/app9183750>
12. Liu J, Wang X (2020) Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network. *Front Plant Sci* 11(6):1–12. <https://doi.org/10.3389/fpls.2020.00898>
13. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021) Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. *Sustainable Cities and Society* 65(2020)
14. Perng DB, Chen YC, Lee MK (2005) A novel AOI system for OLED panel inspection. In *Journal of Physics: Conference Series* (Vol. 13, No. 1, p. 081). IOP Publishing. <https://doi.org/10.1088/1742-6596/13/1/081>
15. Lu RS, Shi YQ, Li Q, Yu QP (2010) AOI techniques for surface defect inspection. In *Applied mechanics and Materials* (Vol. 36, pp. 297–302). Trans Tech Publications Ltd. <https://doi.org/10.4028/www.scientific.net/AMM.36.297>

16. Fan KC, Hsu C (2005) Strategic planning of developing automatic optical inspection (AOI) technologies in Taiwan. In *Journal of Physics: Conference Series* (Vol. 13, No. 1, p. 090). IOP Publishing. <https://doi.org/10.1088/1742-6596/13/1/090>
17. Chen SH, Perng DB (2016) Automatic optical inspection system for IC molding surface. *J Intell Manuf* 27(5):915–926. <https://doi.org/10.1007/s10845-014-0924-5>
18. Acciani G, Brunetti G, Fornarelli G (2006) Application of neural networks in optical inspection and classification of solder joints in surface mount technology. *IEEE Trans Industr Inf* 2(3):200–209. <https://doi.org/10.1109/TII.2006.877265>
19. Richter J, Streitferdt D (2019) Modern architecture for deep learning-based automatic optical inspection. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 141–145). IEEE. <https://doi.org/10.1109/COMPSAC.2019.10197>
20. Liu G, Nouaze JC, Touko Mbouembe PL, Kim JH (2020) YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. *Sensors* 20(7):2145. <https://doi.org/10.3390/s20072145>
21. Shawe-Taylor J, Cristianini N (2004) *Kernel methods for pattern analysis*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511809682>
22. Takeda H, Farsiu S, Milanfar P (2007) Kernel regression for image processing and reconstruction. *IEEE Trans Image Process* 16(2):349–366
23. Li W, Dong R, Fu H, Yu L (2019) Large-scale oil palm tree detection from high-resolution satellite images using two-stage convolutional neural networks. *Remote Sensing* 11(1):11. <https://doi.org/10.3390/rs11010011>
24. Fukushima K (1989) A Hierarchical Neural Network Capable of Visual Pattern Recognition. *Neural Network*. Vol. 1, 1989. [https://doi.org/10.1016/0893-6080\(88\)90014-7](https://doi.org/10.1016/0893-6080(88)90014-7)
25. Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1–6). IEEE. 10.1109/ICEngTechnol. <https://doi.org/2017.8308186>
26. LeCun Y, Bengio Y (1995) Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361(10):1995. <https://doi.org/10.3902/jnns.2.1>
27. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324. <https://doi.org/10.1109/5.726791>
28. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105. <https://doi.org/10.1016/j.aasri.2014.05.013>
29. Bochkovskiy A, Wang CY, Liao HYM (2020) YOLOv4: Optimal speed and accuracy of object detection. *arXiv Preprint arXiv:2004.10934*:1–17. <https://arxiv.org/abs/2004.10934>
30. Ting L, Baijun Z, Yongsheng Z, Shun Y (2021) Ship Detection Algorithm based on Improved YOLO V5. In *2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE)* (pp. 483–487). IEEE. <https://doi.org/10.1109/CACRE52464.2021.9501331>
31. Li D, Tian J (2013) An accurate calibration method for a camera with telecentric lenses. *Opt Lasers Eng* 51(5):538–541. <https://doi.org/10.1016/j.optlaseng.2012.12.008>
32. Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn* 30(7):1145–1159. [https://doi.org/10.1016/S0031-3203\(96\)00142-2](https://doi.org/10.1016/S0031-3203(96)00142-2)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.