



Towards intelligent monitoring system in wire arc additive manufacturing: a surface anomaly detector on a small dataset

Yuxing Li¹ · Haocheng Mu¹ · Joseph Polden¹ · Huijun Li¹ · Lei Wang² · Chunyang Xia³ · Zengxi Pan¹

Received: 18 December 2021 / Accepted: 18 March 2022 / Published online: 28 March 2022
© The Author(s) 2022

Abstract

Rapid developments in artificial intelligence and image processing have presented many new opportunities for defect detection in manufacturing processes. In this work, an intelligent image processing system has been developed to monitor inter-layer deposition quality during a wire arc additive manufacturing (WAAM) process. This system reveals the feasibility and future potential of using computer vision knowledge in WAAM. Information produced from this system is to be used in conjunction with other quality monitoring systems to verify the quality of fabricated components. It is tailored to identify the presence of defects relating to lack of fusion and voids immediately after the deposition of a given layer. The image processing system is built upon the YOLOv3 architecture and through moderate changes on anchor settings and achieves 53% precision on surface anomaly detection and 100% accuracy in identifying the fabricated components' location, providing a prerequisite for high-precision assessment of welding quality. The work presented in this paper presents an inter-layer vision-based defect monitoring system in WAAM and serves to highlight the feasibility of developing such intelligent computer vision systems for monitoring the WAAM process for defects.

Keywords Wire arc additive manufacturing · Intelligent monitoring system · Deep learning · Surface defects detection · YOLOv3

1 Introduction

Wire arc additive manufacturing (WAAM) is a direct energy deposition (DED) process suitable for the fabrication of large metal components due to its high deposition rate and relatively low manufacturing cost [1–3]. WAAM is becoming a popular alternative to traditional manufacturing processes for materials such as the titanium alloys used in aerospace [4, 5] and nickel aluminium bronzes used in marine applications [6]. In the

research domain, work has been published across a number of different aspects of the technology, such as process planning, material property improvements and process control. In pursuit of a fully automated WAAM system, Ding et al. [7] proposed a systematic strategy featuring a series of algorithms, including 2D path planning [8], bead modelling [9] and multi-directional slicing algorithms [10]. The majority of material research tends to focus on material properties for various metals such as titanium alloys [11], aluminium alloys [12], steels [13] and nickel-based alloys [14]. A comprehensive review of material characteristics is provided by Wu et al. [15], Ngo et al. [16] and Liu et al. [17]. Other research efforts investigate methods to improve geometric accuracy [18–20], surface roughness [21] and internal stress build-up [22] in deposited WAAM components. Nevertheless, many challenges remain to be addressed due to the overall complexity of the multifaceted WAAM process [16], such as defect detection, manufacturing process optimization and automatic manufacturing. Recently some studies tracked these research areas: Le et al. [23] optimized the WAAM process parameters, resulting in similar mechanical properties with wrought 308L stainless steel; Ramalho et al. [24] characterized different contaminations through acoustic

✉ Chunyang Xia
yl452@uowmail.edu.au

Zengxi Pan
yl452@uowmail.edu.au

¹ School of Mechanical, Materials, Mechatronic and Biomedical Engineering, University of Wollongong, Wollongong, NSW 2522, Australia

² School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia

³ Institute of Materials Joining, Shandong University, Jinan 250061, Shandong, China

spectrum and achieving quality monitoring; Lopez et al. [25] applied phased array ultrasonic to detect 2- to 5-mm defects in aluminium WAAM component.

Interior defect monitoring is an essential component in ensuring the quality of deposited WAAM components. Although some non-destructive methods, such as X-ray [26, 27], computed tomography [28], ultrasonic testing [29] and eddy current [30], are developed to detect interior defects, but high equipment costs, along with complex detection and analysis processes, restrict the utility of these methods. The visible image captures provide explicit photographs for naked eyes while requiring lower device costs than thermal or infrared cameras. Inspection of visible images also provides clues for deposition quality of a given layer since many internal defects also exhibit visual characteristics at the surface interface. For example, interior voids and lack of fusion defects within the component (two of the most commonly encountered defects associated with WAAM) are often related to the surface quality (smoothness, voids and small area lack of fusion) of the previous layer. These types of defects limit the commercial applications of WAAM, particularly the critical components in naval and aerospace sectors. It is necessary to develop a reliable and intelligent monitoring system for the WAAM processes to predict

potential anomalies in the manufacturing process. These systems will be necessary for the eventual qualification of WAAM components for use in their relevant industries.

In the last decade, deep convolution neural network (CNN) algorithms have pushed to the forefront for image classification and object detection tasks due to their accuracy and speed in extracting static 2D image features. In most scenarios, object detection can be considered for useful than image classification, though it is a more challenging task to compute. Computer vision research communities have invested significant effort into the development of these methods, and they have improved rapidly as a result. Table 1 compares the common object detectors in recent years. Object detectors such as region CNN (R-CNN) [31], single shot detector (SSD) [34], YOLO [36], and their derivatives Faster R-CNN [33], Cascade R-CNN [39], deconvolutional single-shot detector (DSSD) [35] and YOLOv3 [38] have all demonstrated their capabilities in object detection with high accuracy. Researchers in the welding domain have begun applying various algorithms and object detectors to assist technicians in localization tasks and identification of weld defects. Kim et al. [40] applied extraction and noise removal algorithms to identify locations of occluded weld seams from arbitrary

Table 1 Comparisons of different object detectors

	Highlights	Performance	Ref
R-CNN	<ul style="list-style-type: none"> Two-stage object detection algorithm Apply high-capacity convolution neural networks to bottom-up region proposals Supervised pre-training for an auxiliary task with domain-specific fine-tuning returns a significant performance boost 	<ul style="list-style-type: none"> 58.5% mAP on VOC2007 53.3% mAP on VOC2012 	[31]
Fast R-CNN	<ul style="list-style-type: none"> Two-stage object detection algorithm Training is single-stage, using a multi-task loss Training updates all network layers No disk storage required for feature caching 	<ul style="list-style-type: none"> 66.9% mAP on VOC2007 65.7% mAP on VOC2012 	[32]
Faster R-CNN	<ul style="list-style-type: none"> Two-stage object detection algorithm Introduce region proposal network (RPN) for region proposals instead of selective search Merge RPN and Fast R-CNN into a single network by sharing convolutional features-speed boosting 	<ul style="list-style-type: none"> 69.9% mAP on VOC2007 67.0% mAP on VOC2012 	[33]
SSD	<ul style="list-style-type: none"> One-stage object detection algorithm Use pre-set anchor boxes to replace region proposal generation To identify different object sizes, use different feature map to perform detection Suffer from small object, usually less than 36×36 in an image 	<ul style="list-style-type: none"> 79.5% mAP on VOC2007 78.5% mAP on VOC2012 	[34]
DSSD	<ul style="list-style-type: none"> One-stage object detection algorithm Backbone is ResNet-101 instead of VGG19 (SSD) Deconvolution layers are used in detection part to improve the resolution of feature maps 	<ul style="list-style-type: none"> 81.5% mAP on VOC2007 80.0% mAP on VOC2012 	[35]
YOLO	<ul style="list-style-type: none"> Fast object detector good for real-time detection Object information (locations, classes) is obtained from one network Use whole image with pre-set anchors in prediction 	<ul style="list-style-type: none"> 63.4% mAP on VOC2007 57.9% mAP on VOC2012 	[36]
YOLOv2 (YOLO9000)	<ul style="list-style-type: none"> Faster object detector provides a trade-off between speed and accuracy Darknet 19 as backbone in the algorithm Close the gap between detection and classification via WordTree 	<ul style="list-style-type: none"> 78.6% mAP on VOC2007 73.4% mAP on VOC2012 	[37]
YOLOv3	<ul style="list-style-type: none"> Faster object detector, $3.8 \times$ faster than RetinaNet with similar performances Upgrade to Darknet 53 from Darknet 19 	<ul style="list-style-type: none"> 57.9% mAP on COCO 	[38]

directions. Zou et al. [41] proposed a mixed object detector combined with sequence multi-feature combination network (SMFCN) and recurrent neural network (RNN) to detect continuous weld seams from images featuring a significant amount of noise. A simplified YOLOv3 model combined with image augmentation was selected for the task after evaluating performance against faster R-CNN, YOLOv2, and YOLOv3 [42] methods. Another interesting application of YOLOv3 in the welding domain is presented by Dai et al. [43], where positions and evaluated quality of spot welds on an automobile body were accurately detected. The lightweight network MobileNetV3 replaced the backbone network of YOLOv3 achieving a fast detection speed. Based on the previous research work, popular object detectors attained reliable detection results and fast detection speed both in beads location and defect detection in the last several years. Nevertheless, it can be argued that training a defect detector provides more utility than a position detector since defect boundary boxes also provide indication of the location of weld seams [43, 44]. A good defect detector can be used to identify the locations of weld seams with some straightforward optimizations.

The limited dimension of the training dataset is a major factor for the development of intelligent systems on WAAM. Specifically, the large amount of images required for training a deep neural network model is not obtainable in real-world manufacturing scenarios due to incomplete sensor setups and complicated manufacturing environments. To address these challenges, small dataset is widely applied in welding science. Feng et al. [45] used 487 data points to train a deep neural network (DNN) to predict solidification defects, which provides the feasibility to training a DNN model to predict defects with a small dataset. A small X-ray image dataset for weld defects diagnosis achieves more than 90% F1-score in a deep CNN [46]. Therefore, models that apply small datasets are good choices when big datasets are not available in real WAAM manufacturing.

This paper introduces an intelligent model based on the YOLOv3 algorithm. The goal is to utilize a small dataset to identify surface anomalies which occur during the WAAM process. Following this introduction, the paper will introduce the methodology, related important parameters, and algorithms of the YOLOv3. Then three WAAM components featuring a series of defects are fabricated to provide datasets for the training of the model. To test the model, three additional components featuring fewer defects are fabricated and then analysed by the model. To improve the performance of the defect prediction model, some alterations in anchor settings are attempted in order to develop a robustly trained model. The model with the best performance is identified and discussed in the result section, followed by conclusions and a conceptual discussion on future work.

2 Methodology

In the WAAM process, a fabricated component's quality is dependent on the quality of each individual layer. Deposited weld beads are a product of carefully selected process parameters and task-based parameters such as the infill pattern used overlapping distance between weld beads. With recent developments in computer vision, a number of object detectors have demonstrated capabilities which are likely applicable to the WAAM process. In this section, a brief introduction to the classic object detectors is first provided. Then, as a representation of the newer variety of fast and accurate object detectors, YOLOv3 is discussed in the context of defect detection in the WAAM process.

2.1 Related work

R-CNN [31] is a typical two-stage region-based convolution network detector. R-CNN uses a selective search to extract around 2000 region proposals. Each region proposal is warped into a fixed pixel size (227×227) to be compatible with the CNN architecture, which then extracts features for each region proposal. These features are fed into a linear support vector machine (SVM) to identify a specific class, followed by a linear regression model to predict and modify detection windows. R-CNN achieves excellent object detection accuracy based on region proposal. However, each region proposal's features need to be extracted by CNN and saved to disk, resulting in slow detection speed and expensive training cost.

Fast R-CNN [32] is developed to address these drawbacks by sharing computation for proposals. Fast R-CNN also makes use of selective search for region proposals; however, it feeds the whole image and a set of object proposals into the CNN to obtain a convolutional feature map. Then, a pooling layer processes each proposal to extract a corresponding fixed-length feature vector from the feature map. After that, the fixed-sized feature map is transferred to a feature vector and finally branched into two output layers to produce probability estimates and boundary box positions. These changes help to share the convolution result from CNN, avoid repeated computation for region proposals and therefore attain a speed approximately 25 times faster than that of R-CNN. Faster R-CNN [33] focuses on optimizing the method of selecting region proposals. Both R-CNN and fast R-CNN determine region proposal by selective search, while faster R-CNN trains a neural network, defined as region proposal network (RPN), to find all region proposals. Firstly, Faster R-CNN sends input images to the CNN and obtains feature maps. Then, a pre-trained RPN extracts the proposals from

feature maps and unites the proposal information with the feature maps to obtain useful feature information. Linear classifiers and linear regression techniques are then used to predict the presence of select objects. Based on faster R-CNN, many region-based algorithms have been developed to improve the performance and speed of object detection, for instance, Libra R-CNN [47] and Cascade R-CNN [39].

The YOLO series are powerful one-stage object detection algorithms which attempt to balance accuracy and speed. In 2016, YOLO [36] was first published. YOLO [36] divides a 224×224 image into a set of 7×7 grids and then uses a single convolutional network to output a $7 \times 7 \times 30$ tensor. This tensor composes two kinds of boundary box details and the probabilities for different object classes (20 classes in the PASCAL VOC dataset). Box details include x, y coordinates and w, h . The x and y coordinates represent the centre coordinates of the box relative to the bounds of the grid cell. The w and h denote the grid dimensions relative to the dimensions of the whole image. Finally, YOLO predicts whether an object exists in each individual grid. The limitations of YOLO are well known: significant localization errors are compared with fast R-CNN; low recall is compared with two-stage algorithms and difficulty in detecting small objects. YOLOv2 [37] was developed to solve these issues via (amongst other things) batch normalization, a high-resolution classifier and anchor box convolution. In addition, YOLOv2 proposed a new network architecture called Darknet-19, which has 19 convolutional layers and five max-pooling layers. Darknet-19 achieved 91.2% accuracy on ImageNet, which is higher than the 88% achieved by its YOLO backbone network. The output tensor of YOLOv2 is in the size of $13 \times 13 \times 125$ (five prediction boundary boxes, each boundary box has four box parameters, one confidence score and 20 class probabilities). YOLOv3 [38] further improved the prediction performance and small object detection capabilities. The backbone of YOLOv3 is modified to Darknet-53, which features 53 convolution layers and 23 residual blocks. Another innovation of YOLOv3 is using three different scaled output tensors (in the size of 13×13 , 26×26 , and 52×52) to predict different-sized objects by three anchor boxes for each scaled output tensor. YOLOv3 has significant advantages both on accuracy and on speed when the intersection of union (IOU, refer to 2.2.iv) is set to 0.5.

Figure 1 displays a performance comparison between popular object detectors. YOLOv3 has a clear advantage in detection speed while maintaining detection accuracy, which is an essential feature for online real-time detection. Thus, in this paper, we selected YOLOv3 for object detection tasks.

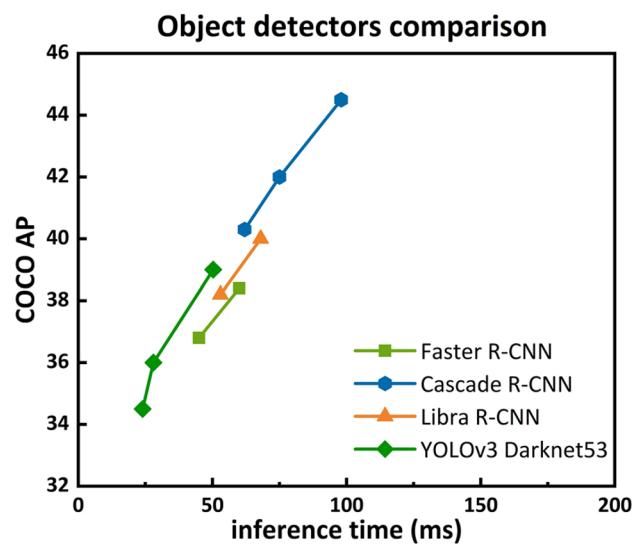


Fig. 1 Comparison of the performance of popular object detectors

2.2 YOLOv3 algorithm

2.2.1 Backbone

Different from its two predecessors (YOLO and YOLOv2), YOLOv3 implements a new network architecture for performing feature extraction. The backbone of YOLOv3, Darknet-53, is a hybrid architecture composed of Darknet-19 [37] and residual block [48]. In Darknet-53, 53 convolutional layers and 23 residual blocks are used to reduce the shape of the input image (416×416) to three feature maps: 13×13 , 26×26 , 52×52 . Then, YOLOv3 uses a feature pyramid network to merge the smaller feature map with the large feature map. Specifically, the feature map produced by Darknet-53 and five convolutional calculations are upsampled and concatenated with the 26×26 feature map. A $26 \times 26 \times 255$ output, including all-class information and boundary box information, is obtained through a series of convolution layers. In a similar fashion, the 26×26 feature map is merged with the 52×52 feature map. Thus, through these merging operations, three different outputs ($13 \times 13 \times 255$, $26 \times 26 \times 255$, $52 \times 52 \times 255$) are obtained. Figure 2 provides the overview of complete YOLOv3 architecture. Darknet-53 is regarded as a backbone to extract image feature maps.

2.2.2 Loss function

YOLOv3 uses mean-squared error and binary cross-entropy to evaluate errors between targets and outputs. Error values for each boundary box width and height use mean-squared error, while other errors including the (x, y) coordinates of the bounding box, confidence scores for objects and non-objects and the probability on class

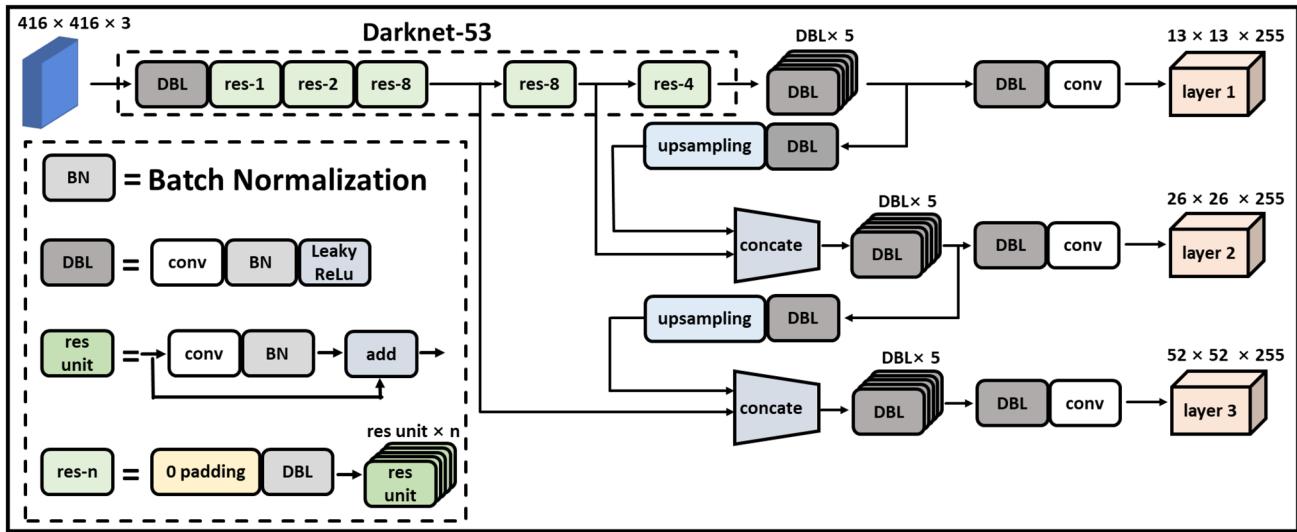


Fig. 2 Architecture of YOLOv3

identification all use binary cross-entropy error [38]. In this paper, in order to facilitate algorithm optimization, we use mean-squared error for box information (x , y , and w , h) and binary cross-entropy error for other parameters. Equation (1) is the loss function used in our training:

$$\begin{aligned}
 loss = & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
 & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[\hat{C}_i \cdot \log C_i + (1 - \hat{C}_i) \cdot \log(1 - C_i) \right] \\
 & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{noobj}} \left[\hat{C}_i \cdot \log C_i + (1 - \hat{C}_i) \cdot \log(1 - C_i) \right] \\
 & + \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{classes}} [\hat{p}_i(c) \cdot \log p_i(c) + (1 - \hat{p}_i(c)) \cdot \log(1 - p_i(c))]
 \end{aligned} \quad (1)$$

The first two terms denote localization loss, terms 3 and 4 represent confidence losses, and the last term indicates the classification loss. λ_{coord} and λ_{noobj} are constants giving weights to the loss terms. We set $\lambda_{\text{coord}} = 1$ and $\lambda_{\text{noobj}} = 100$ in this paper. In YOLOv3, every grid cell receives three prediction boxes based on the sizes of anchor boxes, meaning that B is three. S is the number of grid cells, which is the feature map's shape (for 13×13 feature map, S is 13). 1_{ij}^{obj} and 1_{ij}^{noobj} state if this grid cell is responsible for the object or not. 1_{ij}^{obj} equals 1 if the (i, j) cell contains and be responsible for the object. Otherwise, 1_{ij}^{obj} equals 0. 1_{ij}^{noobj} is 1 if there is no object shown in the grid cell (i, j) or the grid cell is not responsible for the object. Parameters such as x_i , y_i , w_i , h_i , C_i and $p_i(c)$ correspond to x , y coordinates, width and height of prediction boundary box, prediction

confidence, and class probabilities. \hat{x}_i , \hat{y}_i , \hat{w}_i , \hat{h}_i are the ground truth of the labeled boxes. \hat{C}_i and $\hat{p}_i(c)$ are the confidence of the cell's responsibility (which is always 1) and class probability. This loss function tactfully manipulates all losses to a single equation, simplifying the individual optimization processes facing different loss functions.

2.2.3 Anchors

Anchor boxes [33] provide prior knowledge from ground-truth boxes of the training dataset. They deduce the truth box's dimensions through a K-means algorithm. However, instead of Euclidean distance, the intersection of union (IOU) between boundary boxes and centroids is used as the optimization criterion. With this prior knowledge, the trained model predicts the sizes of boundary boxes, improving the model's training convergence rate. The pre-setting of anchor boxes to the model solves the issue of multi-scaled objects. In YOLOv3, nine anchor boxes with different dimensions are pre-set for training and sorted into three groups by size. For instance, three minimum-sized anchor boxes are used as the prior knowledge to predict three boxes for every grid cell in the 13×13 feature map. The predicted boundary box having the largest IOU with the ground-truth box is regarded as the output boundary box. Similar predictions and calculations are designed for 26×26 and 52×52 feature maps, which are designed to precept middle- and small-sized objects, respectively. The nine anchors of YOLOv3 on the COCO dataset [49] are: (10×13) , (16×30) , (33×23) , (30×61) , (62×45) , (59×119) , (116×90) , (156×198) , (373×326) .

2.2.4 Boundary box

Through the pre-set of anchor boxes, every prediction box acquires the best sizes via prior knowledge from anchor boxes. YOLOv3 bonds the anchor boxes' size and prediction boxes' size via Eqs. (2) and (3):

$$b_w = A_w e^{t_w} \quad (2)$$

$$b_h = A_h e^{t_h} \quad (3)$$

where b_w and b_h are the width and height of the final rescaled boundary box. t_w and t_h are the original width and height of the prediction, while A_w and A_h are the scaled anchor box's dimensions. Parameter A_w and A_h are scaled proportionally as the scaled ratio of the feature map (1/32, 1/16, 1/8 for 13 × 13, 26 × 26, 52 × 52 feature maps, respectively). As shown in Fig. 3, the final prediction box's dimensions will be bonded with the anchor box's sizes and scaled into an object area. The centre of the boundary box is calculated by Eqs. (4) and (5):

$$b_x = \sigma(t_x) + c_x \quad (4)$$

$$b_y = \sigma(t_y) + c_y \quad (5)$$

where (t_x, t_y) is the original prediction of the boundary box central coordinates, and (c_x, c_y) is the left corner coordinates of the grid cell. b_x and b_y are the adjusted coordinates of the boundary box centre. As the red point displayed in Fig. 3, the coordinate of the grid cell is $(c_x, c_y) = (3, 2)$. σ is the sigmoid function, which is used to limit the prediction coordinates to absolute coordinates within (0, 1), guaranteeing the centre of the boundary box stays within the grid cell.

As previously described in the ‘Sect. 2.2.3’ section, every grid cell in a feature map receives three boundary boxes of different sizes. After the calculation (Eqs. (2) to (5)), YOLOv3 outputs four coordinate parameters (b_x, b_y, b_w, b_h) and one confidence parameter for each boundary box. To

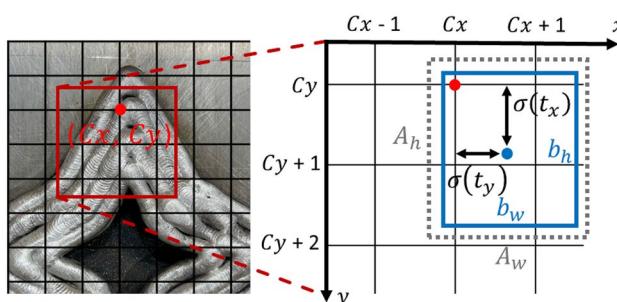


Fig. 3 Illustration of boundary box parameters

select the best and most reliable boundary box, two criteria are designed:

1. Confidence threshold is used to show the confidence that an object exists or not. The boundary box with a confidence lower than the pre-set threshold is discarded.
2. Non-maximum suppression (NMS) threshold: A technique is used to remove redundant boundary boxes and is evaluated by an IOU (Eq. (6)). Boundary boxes having an IOU larger than the NMS threshold will be removed, avoiding using multiple boundary boxes to indicate the same object.

$$\text{IOU} = \frac{\text{intersection area of prediction box and ground truth box}}{\text{union area of prediction box and ground truth box}} \quad (6)$$

3 Experimental result of YOLOv3 on WAAM dataset

3.1 Experimental set-up

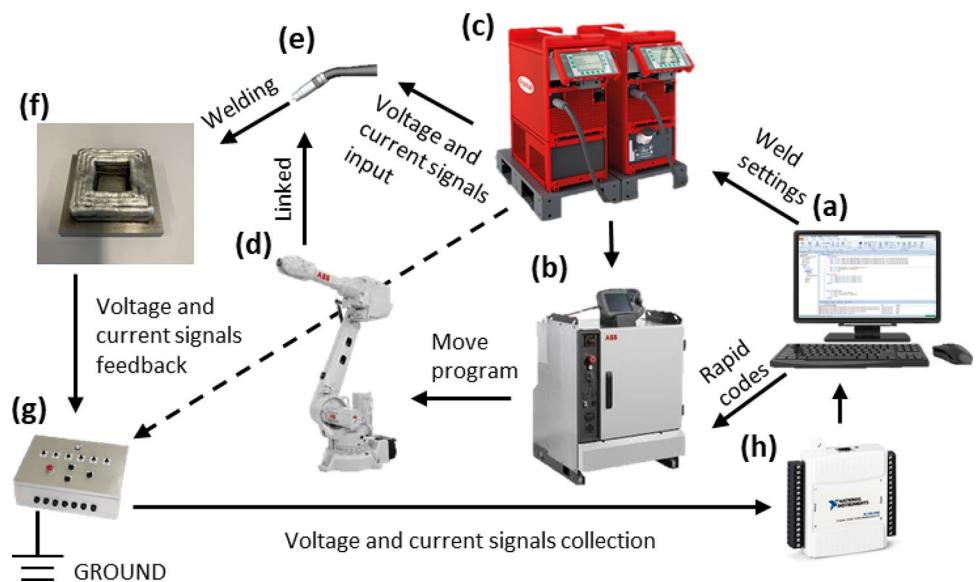
A conventional WAAM system is used to deposit three components for training and then three additional components for testing. The set-up of the WAAM system is shown in Fig. 4. It consists of a control PC (Fig. 4a), an ABB robotic system (Fig. 4b, d), a Fronius welder (Fig. 4c, e), a weld process current and voltage collection system (Fig. 4g), and a national instrument USB 6008 board (Fig. 4h).

The control PC has software that generates code files to control robot movements and synchronize welding parameter settings. The shielding gas consists of 100% Argon. BOC 4043 aluminium MIG wire is used in these tests. Synergic welding control generates weld parameter sets, meaning that the welder automatically determines current and voltage values through pre-stored programs in the Fronius weld controller. In order to get enough defects for the training dataset, the Fronius proprietary cold metal transfer (CMT) welding process is used to fabricate the training components because CMT mode provides lower heat input than the CMT-pulse mode at the same wire feed speed setting. The CMT-pulse process is used in the test components and case study to demonstrate realistic manufacturing scenarios.

3.2 Dataset construction

Three different components, featuring a variety of geometric features, were designed for the training experiments. Three components (square-shaped, hexagon-shaped and star-shaped) are designed in order to generate potential defects when processing right angles, obtuse angles and acutely angled features. Figure 5 provides an illustration of the required welding paths. During

Fig. 4 The set-up of the WAAM system in this experiment. (a) PC controller. (b)(d) ABB robot system: IRC5 controller and IRB2600 robot. (c) (e) Fronius welder and weld torch. (f) actual workpiece. (g) Self-build high-frequency current and voltage collection box. (h) NI USB 6008 board



the manufacturing process, neighbouring contours were welded in clockwise and anticlockwise directions, as displayed in Fig. 5. Figure 5 also highlights the positions where defects are likely to appear. The primary defects that arise at these positions are surface voids because the distance between two vertexes is too large to be covered by the width of welding beads. Lack of fusion between weld beads is another defect encountered in the WAAM process, often generated when the overlap distance between weld paths is too wide.

Fig. 5 Path planning for three components of different shapes: square, hexagon and star

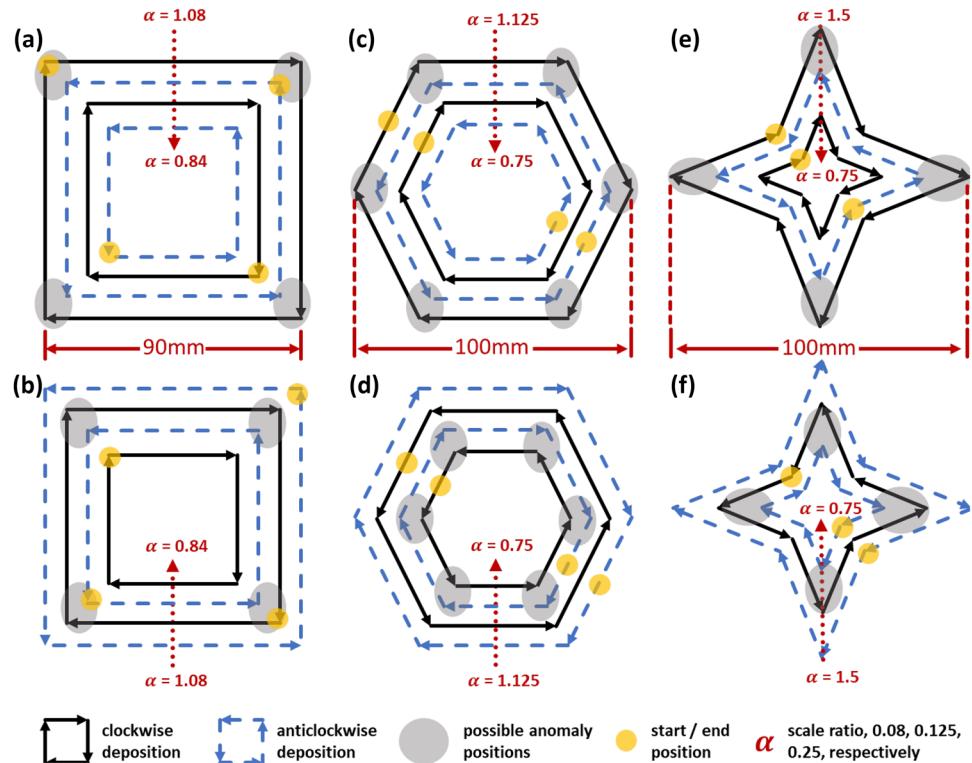


Figure 6 shows the WAAM components after fabrication. During the manufacturing process, after the deposition of each layer, images of the deposited surface are taken with a camera mounted on the welding robot. Three components, each consisting of 30 deposited layers, were built (Fig. 6a–c), and images of each deposited layer were captured and cropped to be used in the training dataset. In total, 51 images are designated for use in the training dataset, whilst the remaining 17 images were used in the validation dataset. Since each

Fig. 6 Three welding components demonstration. From (a) – (c) and (d) – (f) are square, hexagon, and star components, respectively

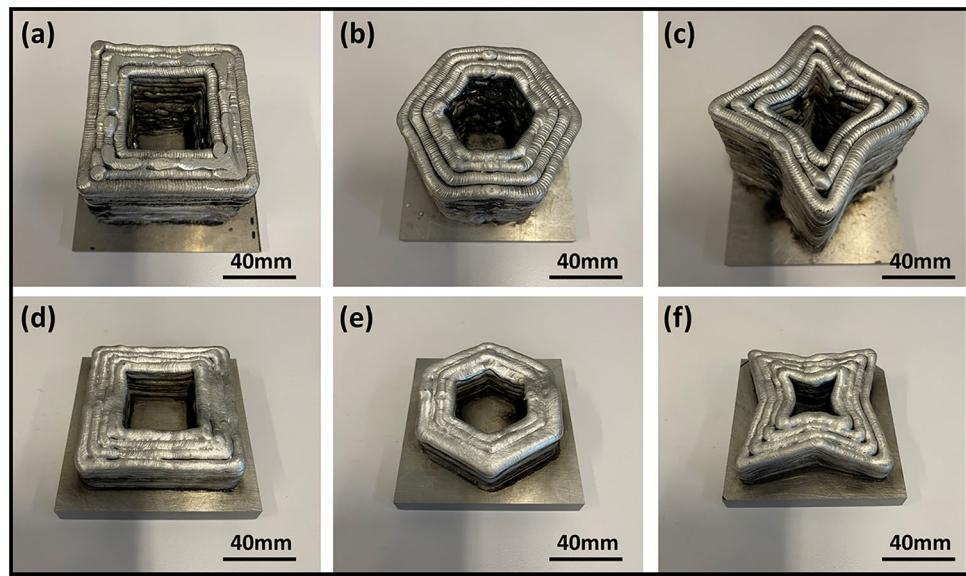


image will contain a different number of anomalies, 1646 anomaly boxes are included in the training dataset and 602 anomaly boxes in the validation dataset. The width between welding beads was purposefully raised to increase the number of defects in the training set components. On the other hand, appropriate wire feed speed and travel speeds were used for the fabrication of components in the test dataset (Fig. 6d–f). Each component in the test dataset consists of 12 welded layers, each of which has several accompanying images to go along with it. In total, 146 images are stored in the test dataset. The welding parameters, wire feed speed (WFS) and travel speed (TS) for each test component are shown in Table 2. Furthermore, each image includes two types of labels: component and anomaly. In general, each image features only one component label.

3.3 Anchors

An Anchor provides prior knowledge about the dimensions of objects. Redmon and Farhadi [38] determined that nine different anchor box dimensions obtain the best performance on the COCO dataset with 80 object classes. However, in our welding dataset, there are only two classes (anomaly and component) in the whole dataset. Furthermore, the dimensions of the anomaly and component boxes are not similar to the objects in the COCO dataset. Thus, new anchor settings are required for our dataset.

K-means algorithm is used to discover clusters along with size information of the ground-truth boundary boxes. The IOU between the anchor boxes is used to optimize cluster centroids. Similar to the anchor sets for the COCO dataset, we keep nine anchors for the training model.

Table 2 Components for test dataset's welding parameters

Layer	Square (exterior to interior order)		Hexagon (exterior to interior order)		Star (exterior to interior order)	
	WFS(m/min)	TS (m/min)	WFS (m/min)	TS (m/min)	WFS (m/min)	TS (m/min)
1	6→5→5→6	0.5	6→5→5→6	0.5	6→5→5→6	0.5
2	6→5→6→6	0.5	6→5→6→6	0.5	6→6→5→6	0.5
3	6→5→5→6	0.5	6→5→5→6	0.5	6→5→5→6	0.5
4	5→5.5→5.5→5	0.5	5→5.5→5.5→5	0.5	5→5.5→5.5→5	0.5
5	5→6→6→5	0.5	5→6→6→5	0.5	5.5→6→6→5.5	0.5
6	5.5→6→6→5.5	0.5	5→5.5→5.5→5	0.5	5.5→6→6→5.5	0.5
7	5.5→6→6→5.5	0.5	5→5.5→5.5→5	0.5	5.5→6→6→5.5	0.5
8	6→6.5→6.5→6	0.5	5.5→6→6→5.5	0.5	6→6.5→6.5→6	0.5
9	5.5→5.5→5.5→5.5	0.5	5→5.5→5.5→5	0.5	5.5→6→6→5.5	0.5
10	5.5→5.5→5.5→5.5	0.5	5→5→5→5	0.5	5.5→6→6→5.5	0.5
11	5→6→6→5	0.5	5→5→5→5	0.5	5.5→6→6→5.5	0.5
12	5.5→6→6→5.5	0.5	5.5→6→6→5.5	0.5	5.5→6→6→5.5	0.5

Table 3 The modified anchor box settings

Anchor	Original anchors	1-anchor-change	2-anchor-change	3-anchor-change	4-anchor-change	9-anchor-change
1	(10, 13)	(10, 13)	(10, 13)	(10, 13)	(22, 14)	(14, 17)
2	(16, 30)	(16, 30)	(21, 20)	(15, 23)	(15, 23)	(23, 12)
3	(33, 23)	(33, 23)	(33, 23)	(26, 15)	(30, 17)	(12, 24)
4	(30, 61)	(30, 61)	(30, 61)	(30, 61)	(30, 61)	(20, 16)
5	(62, 45)	(62, 45)	(62, 45)	(62, 45)	(62, 45)	(17, 21)
6	(59, 119)	(59, 119)	(59, 119)	(59, 119)	(59, 119)	(32, 14)
7	(116, 90)	(116, 90)	(116, 90)	(116, 90)	(116, 90)	(19, 29)
8	(156, 198)	(156, 198)	(156, 198)	(156, 198)	(156, 198)	(28, 20)
9	(373, 326)	(350, 287)	(350, 287)	(350, 287)	(350, 287)	(350, 287)
Group	A	B	C	D	E	F

We modified the original COCO dataset's anchor setting. As the relative size of the anomaly inside the image is small, only the anchor settings for the small object anchors and the largest anchor size (used for component detection) are modified, as displayed in Table 3.

As shown in Table 3, “1-anchor-change” indicates that we only changed the largest anchor size used for component prediction. 2-anchor-change to 4-anchor-change modify anchors for small objects gradually to find out the best anchor setting for our dataset. Finally, 9-anchor-change means that all the ground-truth bounding boxes used during training are clustered into nine classes and completely replace the original anchor settings. To describe the modifications succinctly, we named the six different anchor settings from Group ‘A’ to ‘F’, as outlined in Table 3. It is worth mentioning that to determine the modified anchor box size, boxes in the training dataset are first clustered into nine clusters. Then, the nine anchors without the largest anchor size are clustered into clusters of 1–3, respectively, to obtain the changed anchor data for Group C to Group E.

3.4 Training

The training process is performed in two steps: Firstly, models are trained with only a small number of epochs and the best optimizer is then selected; secondly, the training epoch number is increased so as to compare the models obtained with different anchor settings. The model with the best performance is then selected.

This paper compared the model performances from different optimizers, including Adadelta [50], RMSprop [51], SGD [52] and Adam [53]. To select the optimizer having rapid convergence, a large learning rate and a small epoch number are set in advance, which are 1e-3 and 64, respectively. Then, the best optimizer is trained with a series of small learning rates. Several different random learning rates between 1e-5 and 1e-3 are compared. With the best optimizer and learning rate, we increase the iteration number

to 128 and compare the performances on different anchor settings. Lastly, we find out the best model by comparing the performances in the test dataset. Mean average precision (mAP) is used to evaluate the performance of each model, which is calculated by taking the average precision over all classes. In this study, both component object and anomaly class are calculated. It is defined as below:

$$\text{mAP} = \frac{\sum \text{precision for all classes}}{\text{number of classes}} \quad (8)$$

4 Results

4.1 Training results

To find the best optimizer for our training dataset, the number of epochs is first set to a small number (128), while the learning rate is set to a large value (1e-3). Optimizers tested are the aforementioned Adadelta, RMSprop, SGD and Adam methods. The training processes are shown in Fig. 7. It is found that SGD performs poorly under these hyperparameter settings. The evaluation criteria such as precision and recall are poorly reflected on our dataset. Adadelta was found to outperform RMSprop. Both precision and recall show unstable performances during the training. Figure 8 is the model performance on the validation dataset after each epoch. Based on Fig. 8, generally, the Adam optimizer performs better than the other three optimizers. Therefore, we select Adam as our optimizer in the subsequent experiments. After further tuning the learning rate, we found that the model performs best at the validation dataset when the learning rate is 1.26e-4.

As shown in Fig. 8, the results on the validation dataset improve with increasing epoch number, which suggests that extending beyond 128 will further improve the prediction model. For further models' comparisons, 500 is set for the

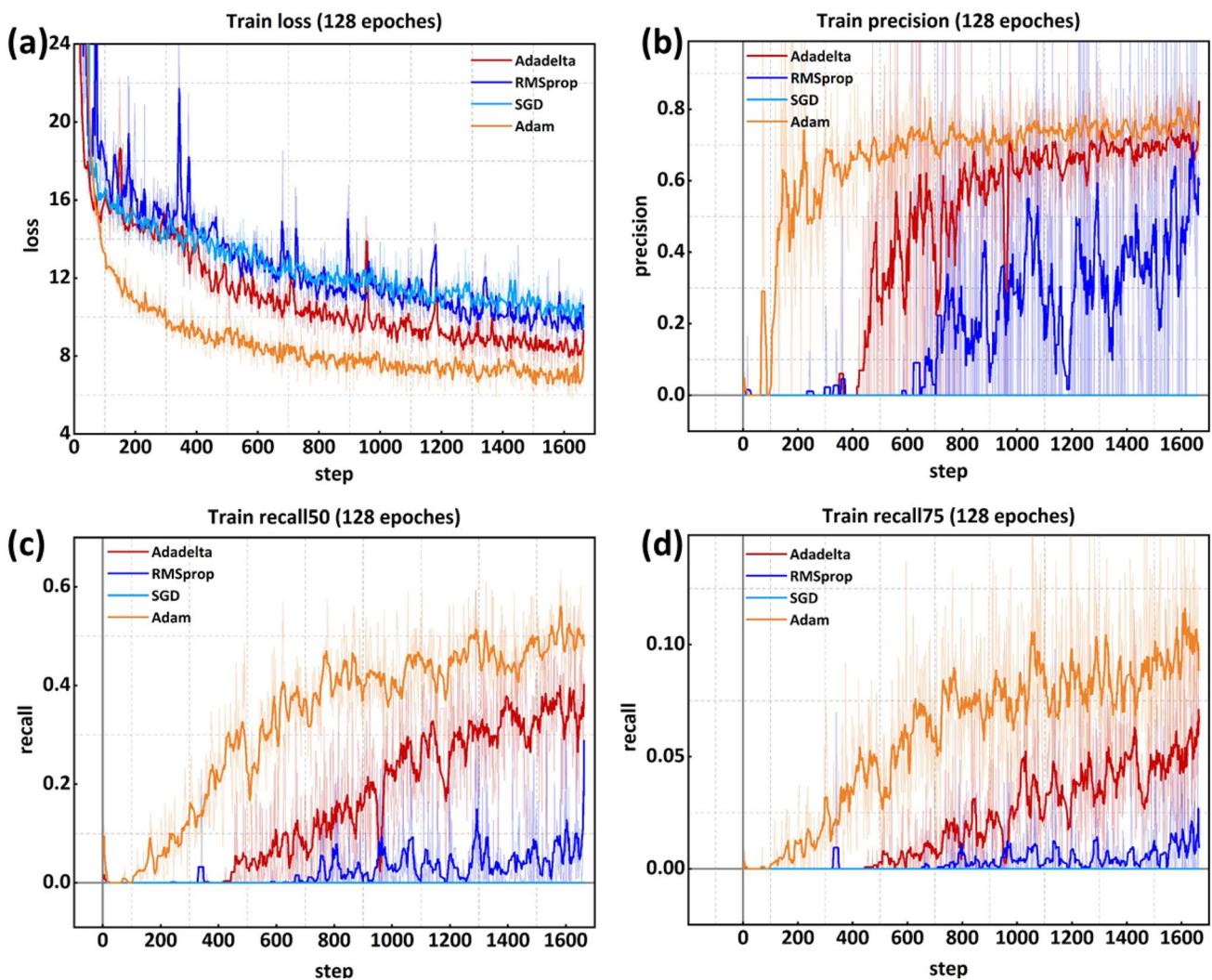


Fig. 7 Training performances: **(a)** total loss; **(b)** precision; **(c)** recall with 0.5 IOU; **(d)** recall with 0.75 IOU on different optimizers: Adadelta, RMSprop, SGD and Adam

number of epochs to achieve reliable results. To determine the best anchor setting, we designed and trained a series of experiments and models to find out the best model for anomaly detection.

Firstly, a model with nine anchors replacement (Group F, as described in Table 3) is trained. For comparison, a model having the original anchor setting (Group A in Table 3) is trained. The performance of the validation dataset is shown in Fig. 9. The model performance for Group A is always better than that of Group F, especially for the F1 score and the mAP criterion. It is mainly because the feature maps from YOLOv3 have three different sizes (13×13 , 26×26 , and 52×52), which are designed to identify objects of various sizes. However, sizes of the boundary boxes of the anomaly are always similar, which are located within the small object ranges. Therefore, if we replace all anchors with small object sizes, the feature maps from YOLOv3 will get confused

when detecting medium- and large-sized objects, performing poorly on our training dataset. Furthermore, the erratic precision for Group F indicates that the model is not stable and reliable for defect prediction. Thus, the anchor setting in Group F cannot be used to detect anomalies in our dataset.

Secondly, the performance of Group A is compared with Groups B, C, D and E. After 500 epochs training, as shown in Fig. 10, Group E is not able to perform as well as Group A, while Groups B, C and D all have better performances than Group A. The comparison results of Group A E and F imply that too many anchor changes reduce the performance and stability of the model. Although Groups B, C and D all have better performance than Group A (based on F1 score and mAP), Group C performs more accurately on the anomaly prediction than the other two groups roughly within 500 epochs. On the other hand, Groups C and D are not as stable as Group B, since evident ups and downs for

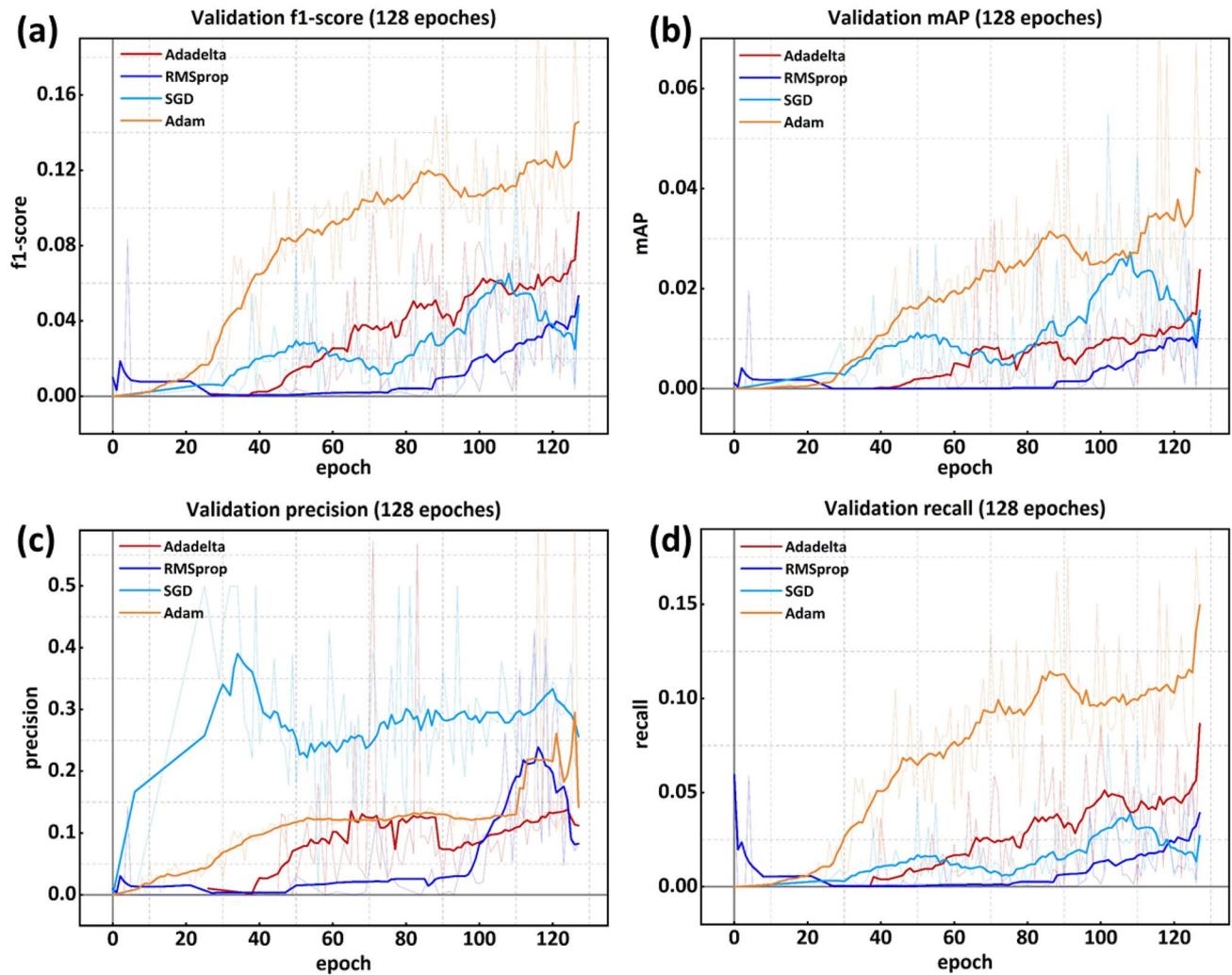


Fig. 8 Validation performances: (a) f1 score; (b) mAP; (c) precision with 0.5 IOU; (d) recall with 0.5 IOU on different optimizers: Adadelta, RMSprop, SGD, and Adam

Groups C and D are presented in Fig. 10. Due to the fluctuation in the performance of Group D, it is hard to compare its performance with Group B.

In conclusion, the model with Group B's anchor setting demonstrates the most stable performance. Anchor settings for Groups C and D have obvious fluctuations. However, within 500 epochs, the performance of Group C is the best compared with Groups B and D. Besides, as shown in Fig. 10, when the epoch number is set to 500, the model's performance improves with the increase in epoch number, especially for the F1 score and mAP criterion. Groups B and D have more obvious tendencies to improve performances at the end of the 500 epoch, while Group C has a faster convergence rate.

As discussed earlier, the best model's performance cannot be obtained within 500 epochs, especially for Groups B and D. Thus, the epoch number is increased to 1000 for training.

The model's performance of anchor settings on Groups B, C and D will be compared directly on images in the test dataset to determine the best anchor setting. The performance of the model with anchor settings for Groups A–D will be displayed and compared in the next section.

4.2 Model performance on test images

With the best learning rate mentioned above (1.26e-4), the four best models on anchor settings of Groups A, B, C and D within 1000 epochs are selected and compared on the test dataset. To find out the correct confidence threshold and NMS threshold intervals, the performance on the validation dataset is compared in advance. Preliminary values for the confidence threshold vary from 0.3 to 1, and the box with confidence smaller than 0.3 is regarded as a negative sample. Through our comparison, high confidence thresholds (larger

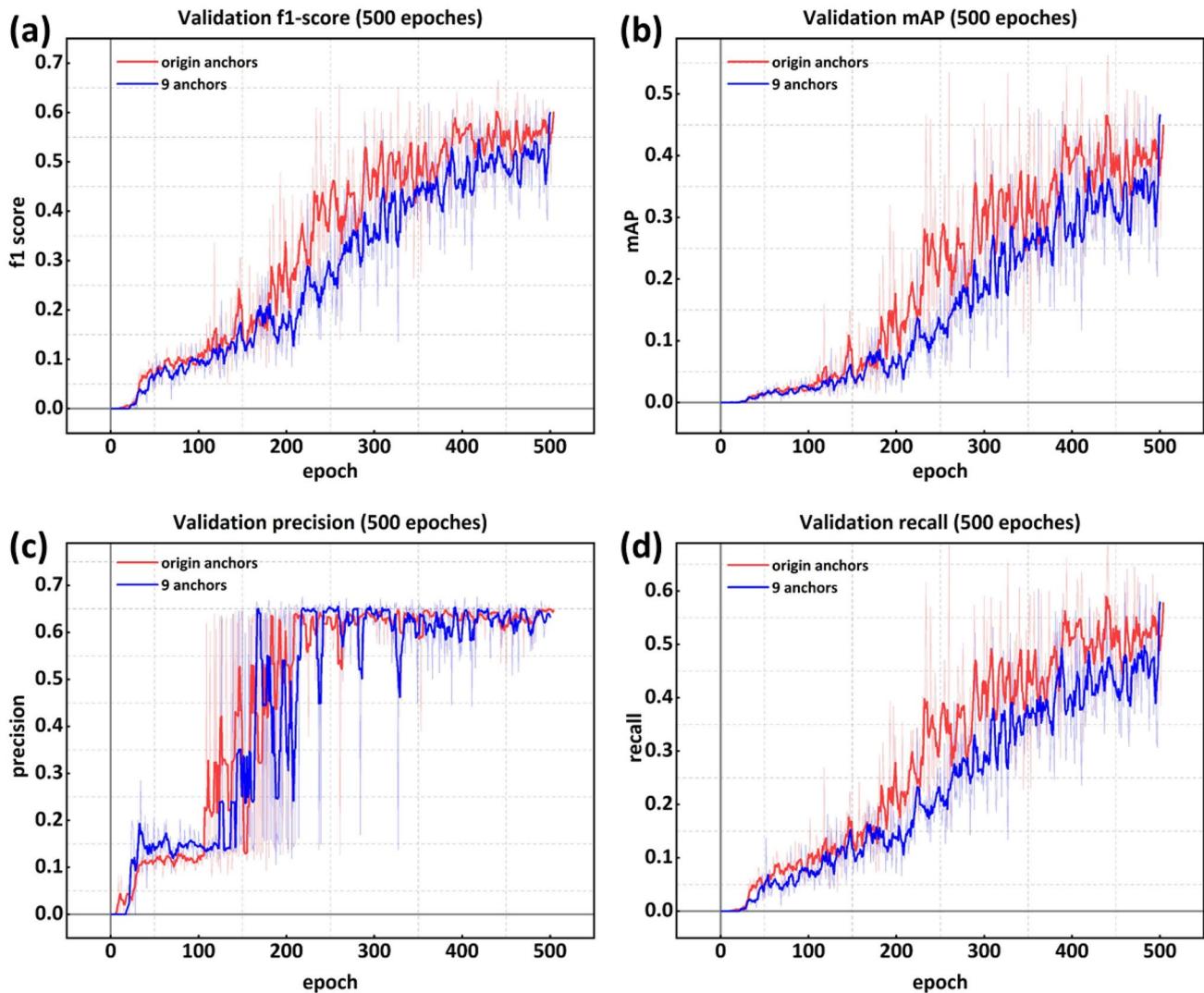


Fig. 9 The model performances with anchor setting on Group A and Group F: (a) F1 score; (b) mAP; (c) precision with 0.5 IOU; (d) recall with 0.5 IOU

than 0.7) fail to detect the existence of anomalies. On the other hand, too large NMS thresholds keep boundary boxes with high intersection areas, resulting in redundant boundary boxes in predictions. Therefore, in this paper, confidence thresholds are set between 0.4 and 0.7, and NMS threshold is set as 0.05.

Table 4 reports the best model's performance when we set the confidence thresholds and the NMS threshold in the determined intervals. NMS threshold of 0.05 indicates that almost all redundant boundary boxes with intersection areas will be removed after the NMS algorithm. The original anchor setting (Group A) performs not as well as the other three groups, which is consistent with the results shown in Fig. 10. Based on the results shown in Table 4, Groups B, C and D improve at least 5% on the mAP with respect to the original anchor setting. With the increase in the number of

iterations (after 500 epochs), the performances of Groups B and D both achieve better results than that of Group C. Consistent with the previous discussion, Group C has a quick convergence rate, resulting in low improvements with the increase in epoch number, while Groups B and D improve their prediction results on both component and anomaly areas with the increased training of epoch number. Comparing with Groups B and C, Group D performs better both on component and anomaly identifications. Group D has a 5.8% improvement on the component identification compared with the original anchor setting (Group A) and an 8.9% improvement on anomaly prediction when the confidence threshold is set to 0.4. There is about a 7.5% improvement on Group D's mAP when the threshold is set at 0.4 and 0.5 compared with Group A. About 2% mAP rises on Group D relative to Group B and C from Table 4.

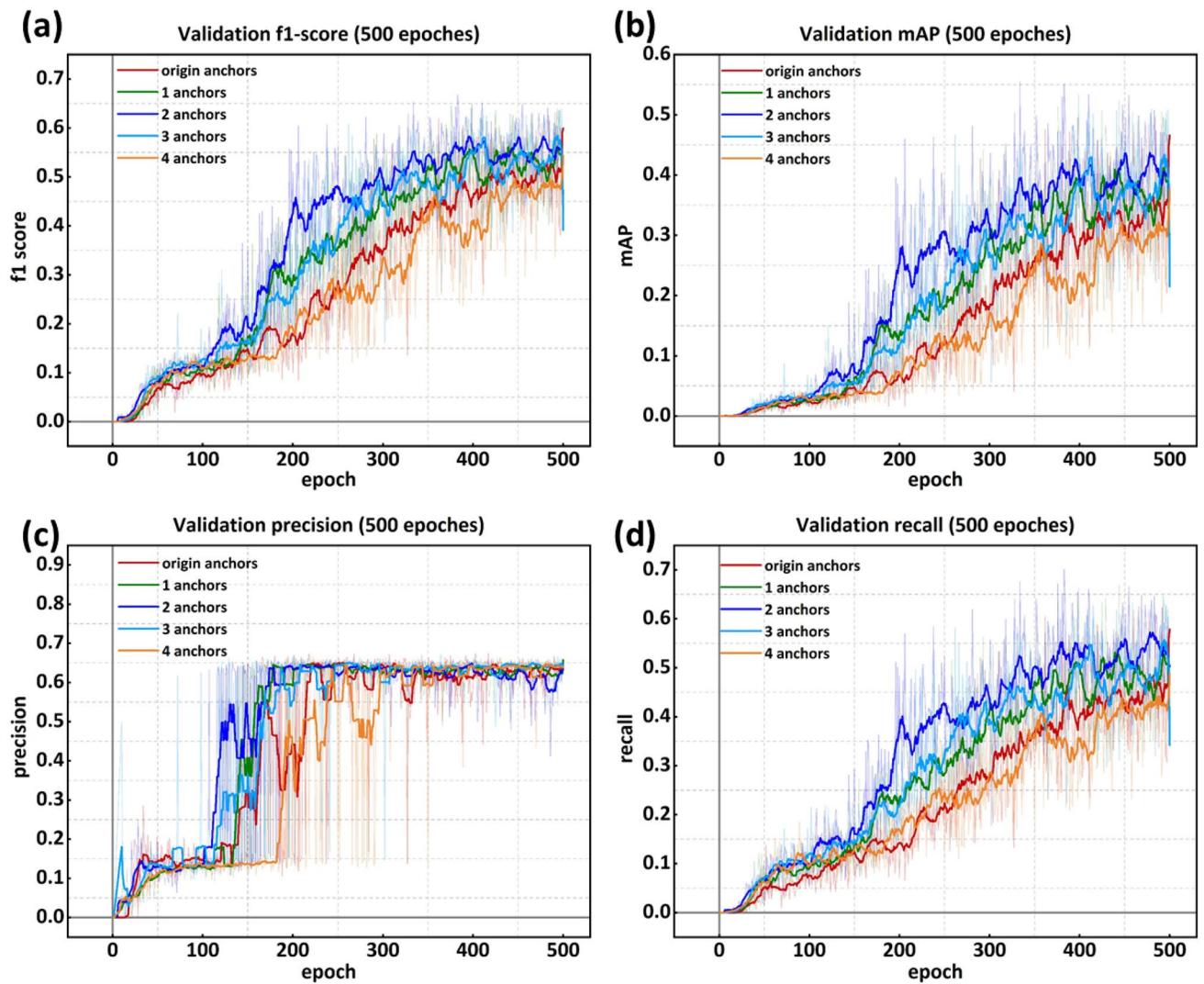


Fig. 10 Model performances on validation dataset of Groups A, B, C, D, E on (a) F1 score; (b) mAP; (c) precision with 0.5 IOU; (d) recall with 0.5 IOU

The four best models are evaluated on the test dataset. Figure 11 displays results for the best models of Groups A–D at a 0.5 confidence threshold. Some obvious undetected anomalies are also highlighted. The best model of the original anchor setting (Group A) performs appropriately on both anomaly and component. However, the accuracy of anomaly detection and the size of boundary boxes are the main issues of the original anchor setting. Some anomalies are not detected in Fig. 11a–c, resulting in the drop of mAP. Prior knowledge of anchor size is not used in this group, bringing inconsistent anomaly boundary box sizes (as shown in Fig. 11a–c). The performance results are consistent with the evaluation data in Table 4: The models from Groups B, C and D identify component boxes in the right positions, and the performance of anomaly detection on Group D is the best, followed by Group B and then Group C. However,

it is worth mentioning that the boundary boxes in Groups B and C (Fig. 11d–i) have acceptable dimensions for the component prediction, while the boundary boxes in Group D are too large for components. An abrupt and significant drop (23.5% from Table 4) of component identification accuracy happens when the confidence threshold increases from 0.5 to 0.6, indicating that the use of larger box sizes is improper.

Next, models from Groups B and C are candidate models for future applications. Further identification results on the test dataset are shown in Fig. 12. For the model with only 1-anchor-change, the prediction for anomalies is acceptable when the confidence threshold is 0.4 (Fig. 12a–c). Most of the unfused parts are identified and marked within boundary boxes (around 53% accuracy). The model with a 0.5 confidence threshold (Fig. 12d–f) removes some low confidence boundary boxes, resulting

Table 4 The four best models (at epoch 988, 795, 942, 947, respectively) performance on the validation dataset

	Confidence threshold	NMS threshold	Component	Anomaly	mAP
Origin anchors	0.4	0.05	0.9412	0.4836	0.7124
(Group A)	0.5	0.05	0.9412	0.4794	0.7103
	0.6	0.05	0.7647	0.4703	0.6175
	0.7	0.05	0.5294	0.4541	0.4918
1-anchor-change	0.4	0.05	1.0000	0.5299	0.7650
(Group B)	0.5	0.05	1.0000	0.5273	0.7637
	0.6	0.05	0.8235	0.5157	0.6696
	0.7	0.05	0.7059	0.4848	0.5954
2-anchor-change	0.4	0.05	1.0000	0.5264	0.7632
(Group C)	0.5	0.05	1.0000	0.5157	0.7579
	0.6	0.05	0.9412	0.4990	0.7201
	0.7	0.05	0.7647	0.4889	0.6268
3-anchor-change	0.4	0.05	1.0000	0.5728	0.7864
(Group D)	0.5	0.05	1.0000	0.5723	0.7862
	0.6	0.05	0.7647	0.5622	0.6635
	0.7	0.05	0.6471	0.5432	0.5952

Fig. 11 Best model's performance on three test components images at 0.5 confidence threshold, respectively, (a)–(c): original anchor setting (Group A); (d)–(f): 1-anchor-change (Group B); (g)–(i): 2-anchor-change (Group C); (j)–(l): 3-anchor-change (Group D)

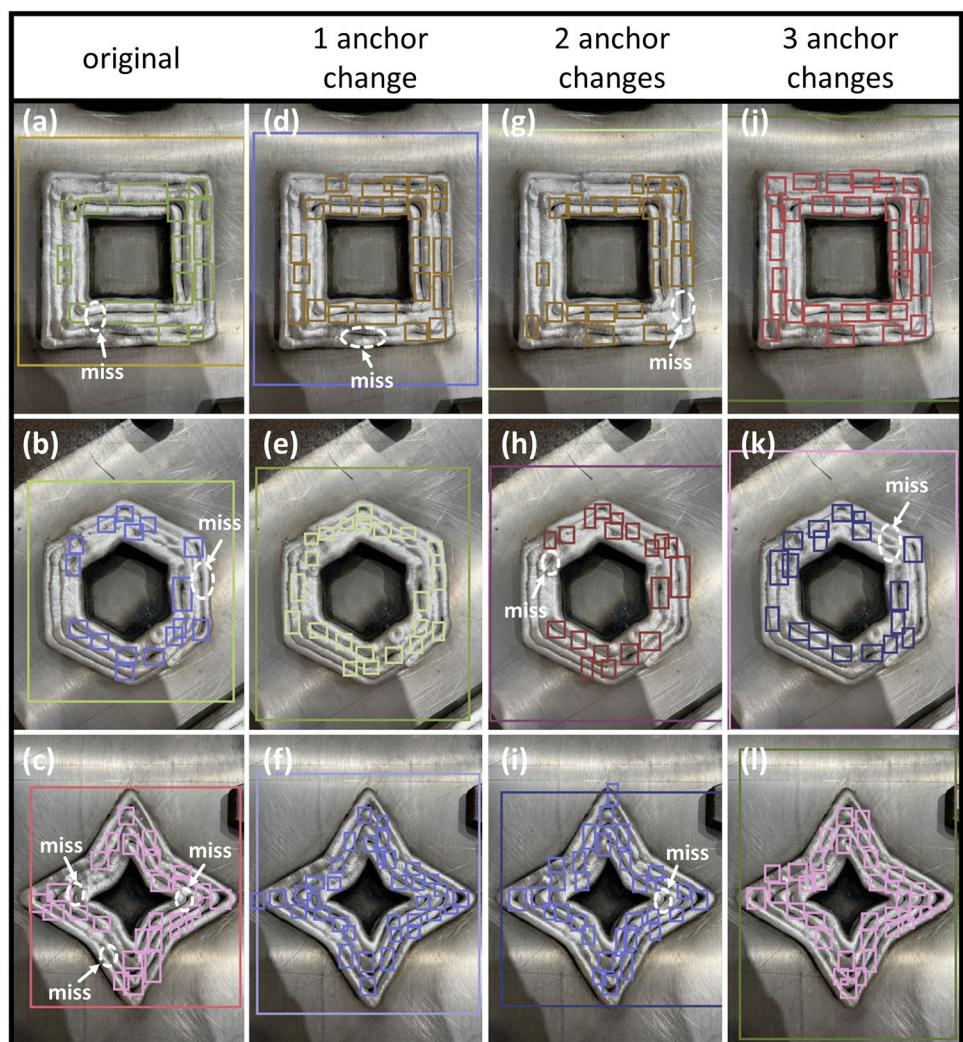


Fig. 12 Performances comparisons on 1-anchor-change and 2-anchor-change: (a)–(c): 1-anchor-change at confidence threshold 0.4; (d)–(f): 1-anchor-change at confidence threshold 0.5; (g)–(i): 2-anchor-changes at confidence threshold 0.4; (j)–(l): 2-anchor-change at confidence threshold 0.5

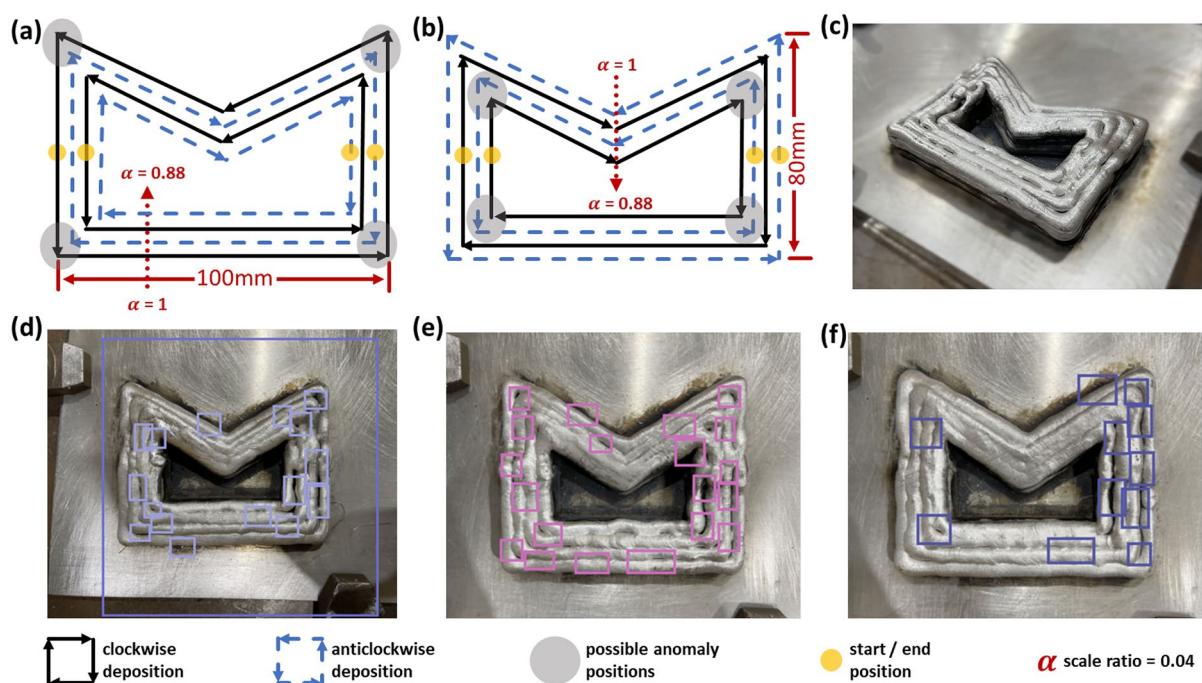
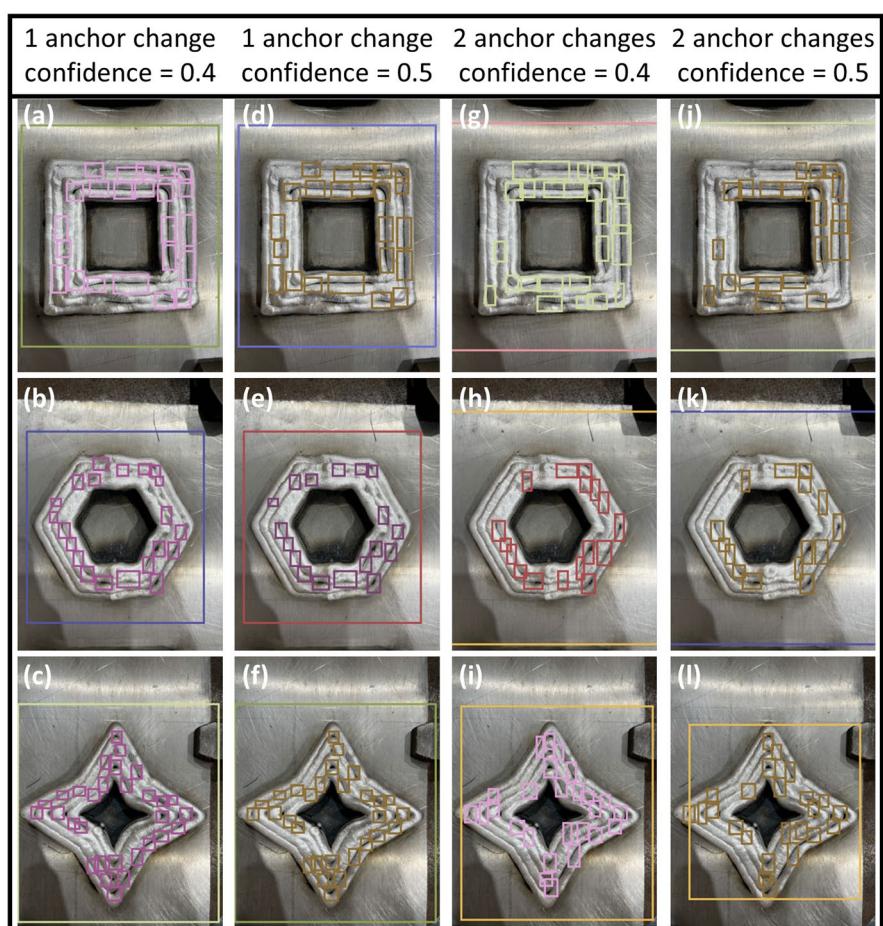


Fig. 13 Illustrations of the component and the detection results: (a)–(b): Path plannings and component dimensions. (c) Real photograph after eight-layer manufacture. (d)–(f): The detection results using the pre-trained best 1-anchor-change model

in missing some real anomaly parts. Similar results for 2-anchor-change models are shown in Fig. 12g–l. The high confidence threshold deletes some boxes with right anomalies inside. When comparing with 1-anchor-change model and 2-anchor-change model, it is noticed that the sizes of the component boxes are always larger in the 2-anchor-changes model than the 1-anchor-change model, even exceeding the size of the image sizes.

Therefore, the model with 1-anchor-change and a 0.4 confidence threshold is selected lastly as the best prediction model for anomalies and component identifications. Based on the data in Table 4, the precision on component prediction achieves 100% precision, while anomaly prediction receives 53% accuracy, with a 76.5% mAP globally.

5 Case study

A component featuring right angles, obtuse angles and acute angles was fabricated to test the model performance in a case study replicating a real-world manufacturing scenario. The component, shown in Fig. 13, is comprised of eight layers, and welding parameters were kept constant throughout the manufacturing process: wire feed rate at 5.5 m/min and travel speed of the robot is 0.5 m/min. The shielding gas used in this case study is 100% argon. The welding process used for this component is Fronius' CMT-pulse. Figure 13 also provides illustrations of the component structure, path planning layout and prediction results. In similar fashion to previously fabricated components, this component is manufactured in clockwise and anti-clockwise commutatively, as shown in Fig. 13a, b.

As discussed in Sect. 4, the 1-anchor-change model performs best for defect detection on our dataset and therefore is used to determine the performance for different shapes under 0.4 confidence. Figure 13d–f is the prediction result from the model. Although the shape of the component differs from our training dataset, most of the lack-of-fusion defects are detected accurately by the model. However, some defects still go undetected. For example, in Fig. 13e, f, the model misjudges component edges with lack-of-fusion defects on a number of occasions. However, for our application, a model that occasionally predicts normal part as defective is always better than a model that misses possible defects, which reduces the impact of these errors. For the detection of the component, the model produces a number of inaccurate predictions. The model only recognizes the component in Fig. 13d, indicating that the model can identify components of different shapes with low accuracy. The main reason causing this low accuracy in component identification is the small training dataset, especially for the component samples (less than 1000).

6 Conclusion and future work

In this work, the YOLOv3 algorithm is applied to a WAAM process to detect: a) the position of the welded component and b) the existence of welding defects on the welded surface. This paper investigates YOLOv3's performance with different anchor settings with a small training dataset. The best model performance based on the validation dataset is selected for the component position identification and detection of defects. It was found that the original anchor setting for the COCO dataset is unsuitable for the WAAM defect dataset. Moderate changes to the anchor settings (1–2 changes) bring the prior knowledge of the component and anomaly sizes to prediction models, improving the prediction mAP by about 5%. In this paper, the 1-anchor-change model achieves 100% accuracy on component identification and 53% precision on the anomaly prediction. This model proves the feasibility of using a small dataset to train a YOLOv3 model to identify defects in a WAAM process. Furthermore, the accurate identification of component locations provides a prerequisite for high-precision assessment of WAAM components qualities.

The work presented in this paper forms an essential component in a wider intelligent monitoring system for WAAM. It will provide image-based defect detection capabilities that can be integrated with other defect detection subsystems, such as the electrical signals' monitoring system presented in [54]. Future work will focus on embedding the model with a functional graphical user interface (GUI), which will assist technicians in the practical task of detecting the presence of anomalies. An interactive GUI will also provide a more streamlined means of integrating these models into the wider defect detection monitoring system.

Authors' contributions Yuxing Li did conceptualization, methodology and writing—original draft; Haochen Mu and Chunyang Xia performed writing, revising—review and editing; Joseph Polden was involved in writing—review and editing; Huijun Li and Lei Wang supervised the study; Zengxi Pan contributed to supervision and writing—review and editing.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes

were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Ke WC et al (2022) Multi-layer deposition mechanism in ultra high-frequency pulsed wire arc additive manufacturing (WAAM) of NiTi shape memory alloys. *Addit Manuf* 50. <https://doi.org/10.1016/j.addma.2021.102513>
- Lopes JG, Machado CM, Duarte VR, Rodrigues TA, Santos TG, Oliveira JP (2020) Effect of milling parameters on HSLA steel parts produced by Wire and Arc Additive Manufacturing (WAAM). *J Manuf Process* 59:739–749. <https://doi.org/10.1016/j.jmapro.2020.10.007>
- Rodrigues TA et al (2021) Effect of heat treatments on 316 stainless steel parts fabricated by wire and arc additive manufacturing : Microstructure and synchrotron X-ray diffraction analysis. *Addit Manuf* 48. <https://doi.org/10.1016/j.addma.2021.102428>
- Williams SW, Martina F, Colegrove PA (2013) Improved microstructure and increased mechanical properties of additive manufacture produced Ti-6Al-4V by interpass cold rolling.
- Williams SW, Martina F, Addison AC, Ding J, Pardal G, Colegrove P (2016) Wire + arc additive manufacturing. *Mater Sci Technol* 32(7):641–647. <https://doi.org/10.1179/1743284715y.0000000073>
- Ding D, Pan Z, van Duin S, Li H, Shen C (2016) Fabricating superior NiAl bronze components through wire arc additive manufacturing. *Materials* 9(8). <https://doi.org/10.3390/ma9080652>
- Ding D et al (2016) Towards an automated robotic arc-welding-based additive manufacturing system from CAD to finished part. *Comput Aided Des* 73:66–75. <https://doi.org/10.1016/j.cad.2015.12.003>
- Ding D, Pan Z, Cuiuri D, Li H (2015) A practical path planning methodology for wire and arc additive manufacturing of thin-walled structures. *Robot Comput Integr Manuf* 34:8–19. <https://doi.org/10.1016/j.rcim.2015.01.003>
- Ding D, Pan Z, Cuiuri D, Li H, van Duin S, Larkin N (2016) Bead modelling and implementation of adaptive MAT path in wire and arc additive manufacturing. *Robot Comput Integr Manuf* 39:32–42. <https://doi.org/10.1016/j.rcim.2015.12.004>
- Ding D, Pan Z, Cuiuri D, Li H, Larkin N, van Duin S (2016) Automatic multi-direction slicing algorithms for wire based additive manufacturing. *Robot Comput Integr Manuf* 37:139–150. <https://doi.org/10.1016/j.rcim.2015.09.002>
- Kim TB, Yue S, Zhang Z, Jones E, Jones JR, Lee PD (2014) Additive manufactured porous titanium structures: Through-process quantification of pore and strut networks. *J Mater Process Technol* 214(11):2706–2715. <https://doi.org/10.1016/j.jmatprotec.2014.05.006>
- Wang P, Hu S, Shen J, Liang Y (2017) Characterization the contribution and limitation of the characteristic processing parameters in cold metal transfer deposition of an Al alloy. *J Mater Process Technol* 245:122–133. <https://doi.org/10.1016/j.jmatprotec.2017.02.019>
- Bissacco G, Hansen HN, De Chiffre L (2005) Micromilling of hardened tool steel for mould making applications. *J Mater Process Technol* 167(2–3):201–207. <https://doi.org/10.1016/j.jmatprotec.2005.05.029>
- Baufeld B (2011) Mechanical properties of INCONEL 718 parts manufactured by shaped metal deposition (SMD). *J Mater Eng Perform* 21(7):1416–1421. <https://doi.org/10.1007/s11665-011-0009-y>
- Wu B et al (2018) A review of the wire arc additive manufacturing of metals: properties, defects and quality improvement. *J Manuf Process* 35:127–139. <https://doi.org/10.1016/j.jmapro.2018.08.001>
- Ngo TD, Kashani A, Imbalzano G, Nguyen KTQ, Hui D (2018) Additive manufacturing (3D printing): A review of materials, methods, applications and challenges. *Compos B Eng* 143:172–196. <https://doi.org/10.1016/j.compositesb.2018.02.012>
- Liu J, Xu Y, Ge Y, Hou Z, Chen S (2020) Wire and arc additive manufacturing of metal components: a review of recent research developments. *Int J Adv Manuf Technol* 1–50
- Xiong J, Zhang G, Hu J, Wu L (2012) Bead geometry prediction for robotic GMAW-based rapid manufacturing through a neural network and a second-order regression analysis. *J Intell Manuf* 25(1):157–163. <https://doi.org/10.1007/s10845-012-0682-1>
- Xiong J, Zhang G (2014) Adaptive control of deposited height in GMAW-based layer additive manufacturing. *J Mater Process Technol* 214(4):962–968. <https://doi.org/10.1016/j.jmatprotec.2013.11.014>
- Xiong J, Yin Z, Zhang W (2016) Closed-loop control of variable layer width for thin-walled parts in wire and arc additive manufacturing. *J Mater Process Technol* 233:100–106. <https://doi.org/10.1016/j.jmatprotec.2016.02.021>
- Xiong J, Li YJ, Yin ZQ, Chen H (2018) Determination of surface roughness in wire and arc additive manufacturing based on laser vision sensing. *Chin J Mech Eng* 31(1). <https://doi.org/10.1186/s10033-018-0276-8>
- Li R, Xiong J, Lei Y (2019) Investigation on thermal stress evolution induced by wire and arc additive manufacturing for circular thin-walled parts. *J Manuf Process* 40:59–67. <https://doi.org/10.1016/j.jmapro.2019.03.006>
- Le VT, Mai DS, Doan TK, Paris H (2021) Wire and arc additive manufacturing of 308L stainless steel components: Optimization of processing parameters and material properties. *Eng Sci Technol Int J* 24(4):1015–1026. <https://doi.org/10.1016/j.jestch.2021.01.009>
- Ramalho A, Santos TG, Bevans B, Smoqi Z, Rao P, Oliveira JP (2022) Effect of contaminations on the acoustic emissions during wire and arc additive manufacturing of 316L stainless steel. *Addit Manuf* 51. <https://doi.org/10.1016/j.addma.2021.102585>
- Lopez AB, Santos J, Sousa JP, Santos TG, Quintino L (2019) Phased Array Ultrasonic Inspection of Metal Additive Manufacturing Parts. *J Nondestruct Eval* 38(3). <https://doi.org/10.1007/s10921-019-0600-y>
- Babot D, Berodias G, Peix G (1991) Detection and sizing by X-ray Compton scattering of near-surface cracks under weld deposited cladding. *NDT E Int* 24(5):247–251
- Naito S, Yamamoto S, Yamamoto S (2009) Novel X-ray backscatter technique for detecting crack below deposit. Toshiba Corporation, Japan. <http://www.ndt.net/article/jrc-nde2009/papers/110.pdf>
- Mireles J, Ridwan S, Morton PA, Hinojos A, Wicker RB (2015) Analysis and correction of defects within parts fabricated using powder bed fusion technology. *Surf Topogr Metrol Prop* 3(3). <https://doi.org/10.1088/2051-672x/3/3/034002>
- Cerniglia D, Scafidi M, Pantano A, Rudlin J (2015) Inspection of additive-manufactured layered components. *Ultrasonics* 62:292–298. <https://doi.org/10.1016/j.ultras.2015.06.001>
- Du W, Bai Q, Wang Y, Zhang B (2017) Eddy current detection of subsurface defects for additive/subtractive hybrid manufacturing. *Int J Adv Manuf Technol* 95(9–12):3185–3195. <https://doi.org/10.1007/s00170-017-1354-2>
- Girshick R, Donahue J, Darrell T, Malik J (2014) RCNN: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 580–587

32. Girshick R (2015) Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision 1440–1448
33. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv Neural Inf Process Syst* 91–99
34. Liu W et al (2016) SSD: Single shot multibox detector. European conference on computer vision. Springer, pp 21–37
35. Fu CY, Liu W, Ranga A, Tyagi A, Berg AC (2017) Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659
36. 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 779–788
37. Redmon J, Farhadi A (2017) YOLOV2:YOLO9000: better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 7263–7271
38. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767
39. Cai Z, Vasconcelos N (2018) Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 6154–6162
40. Kim J, Lee J, Chung M, Shin Y-G (2020) Multiple weld seam extraction from RGB-depth images for automatic robotic welding via point cloud registration. *Multimed Tools Appl* 80(6):9703–9719. <https://doi.org/10.1007/s11042-020-10138-7>
41. Zou Y, Zhu M, Chen X (2021) A robust detector for automated welding seam tracking system. *J Dyn Syst Meas Control* 143(7). <https://doi.org/10.1115/1.4049547>
42. Yang L, Fan J, Liu Y, Li E, Peng J, Liang Z (2021) Automatic detection and location of weld beads with deep convolutional neural networks. *IEEE Trans Instrum Meas* 70:1–12. <https://doi.org/10.1109/tim.2020.3026514>
43. Dai W et al (2021) Deep learning assisted vision inspection of resistance spot welds. *J Manuf Process* 62:262–274. <https://doi.org/10.1016/j.jmapro.2020.12.015>
44. Geng H, Li J, Xiong J, Lin X, Zhang F (2017) Optimization of wire feed for GTAW based additive manufacturing. *J Mater Process Technol* 243:40–47. <https://doi.org/10.1016/j.jmatprotec.2016.11.027>
45. Feng S, Zhou H, Dong H (2019) Using deep neural network with small dataset to predict material defects. *Mater Des* 162:300–310. <https://doi.org/10.1016/j.matdes.2018.11.060>
46. Ajmi C, Zapata J, Martínez-Álvarez JJ, Doménech G, Ruiz R (2020) Using deep learning for defect classification on a small weld x-ray image dataset. *J Nondestruct Eval* 39(3). <https://doi.org/10.1007/s10921-020-00719-9>
47. Pang J, Chen K, Shi J, Feng H, Ouyang W, Lin D (2019) Libra r-cnn: Towards balanced learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 821–830
48. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 770–778
49. Lin T-Y et al (2014) Microsoft coco: Common objects in context. European conference on computer vision. Springer, pp 740–755
50. Zeiler MD (2012) Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701
51. Hinton G, Srivastava N, Swersky K (2012) Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on 14(8):2
52. Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010: Springer 177–186
53. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
54. Li Y et al. (2021) A defect detection system for wire arc additive manufacturing using incremental learning. *J Ind Inf Integr.* <https://doi.org/10.1016/j.jii.2021.100291>

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