CTU Journal of Innovation and Sustainable Development
ISSN 2588-1418 | e-ISSN 2815-6412

DOI:10.22144/ctujoisd.2026.030

Reconstruction of optimal weld seam trajectory for three axis robot using deep learning and quadratic regression approaches

Nguyen Duc Tai[1], Le Viet Phuong[2], To Thanh Tuan[3], Truong Quoc Bao[1], and Nguyen Hoang Dung[1*]

[1]*Faculty of Automation Engineering, Can Tho University, Viet Nam*

[2]*Faculty of Information Technology, An Giang Vocational College, Viet Nam*

[3]*Faculty of Weld Engineering, Lilama 2 International College, Viet Nam*

**Corresponding author (hoangdung@ctu.edu.vn)*

Article info.

Received 1 Aug 2025
Revised 5 Sep 2025
Accepted 27 Feb 2026

Keywords

Linear regression, quadratic regression, weld seam trajectory, weld segmentation, YOLOv8

ABSTRACT

This study proposes a method for the integration of a segmentation model with non-linear regression to construct weld-seam trajectories. First, the weld is precisely segmented using the YOLOv8 model. Then, its boundary pixels are extracted right after and termed SEG points. The 48, 24, and 16 SEG points are utilized as the input for regression models to estimate the weld seam center-line. Both linear and non-linear (quadratic) regression models are assessed using different types of weld images. The experimental results show that the average segmentation performance and training accuracy of the YOLO model are 94%. Additionally, both linear- and nonlinear regression models can estimate similar weld seam profiles. It is worth noting that the more the SEG points are utilized, the higher the accuracy is. However, if a number of SEG points increase, the processing time increases too. Therefore, 24 SEG points are enough for the precise estimation of the weld seam trajectory. It demonstrated that the proposed approach achieves a mean absolute trajectory error of 0.24 mm at 2.9 FPS for the end-to-end pipeline by using the Ampere GPU of NVIDIA Jetson Nano (input 416×416, FP16). Moreover, estimated weld seam trajectory is transformed into the absolute coordinate for the robot's movement.

1. INTRODUCTION

For current industrial welding, almost all quality inspection of welds for compliance with standards is based on manual methods. It takes time and often leads to errors caused by human error. Therefore, this work proposes an automatic welding inspection solution based on image processing and a deep learning network. To inspect the weld seams automatically, a robot arm equipped with a camera moves and follows the weld seams. Therefore, reconstruction of the weld seam is necessary to ensure the robot precisely tracks the expected trajectory. And the weld seam is reconstructed using a nonlinear regression method.

Deep learning (DL), a subfield of machine learning, supports segmentation, detection, classification, and regression using deep neural networks. The DL based approaches such as principal component analysis (Mirapeix et al., 2007), finite element algorithm (Muhammad et al., 2024), and fuzzy logic (Nguyen et al., 2024) have been widely applied in varieties of different fields (Sarker, 2021) including non-destructive weld inspection methods (Oh et al., 2020; Ding et al., 2022; Penekalapati et al., 2022;

Kumaresan et al., 2023; Zhang & Ni, 2023; Sun et al., 2024; Xu & Li, 2024; Yuan et al., 2024; Zhou et al., 2024; Lappin et al., 2025) and weld seam trajectories (Li et al., 2017; Zhao et al., 2024; Zou & Wan, 2024; Qadoori et al., 2025).

Metal welding is widely applied in both daily life and industry. In particular, to ensure weld integrity in high-precision and safety-critical sectors such as aerospace, shipbuilding, heavy mechanical fabrication, and steel structure construction, weld performance plays an extremely important role (Hashim et al., 2024). For these high-tech fields, a single latent weld defect can compromise structural reliability and cause serious consequences, including costly downtime (Deepak et al., 2021).

There are several modern technologies applied to weld detection. Traditional non-destructive testing (NDT) techniques, including ultrasonic testing (UT), radiographic testing (RT), and liquid-penetrant inspection are the workhorses for subsurface flaw detection (Jolly et al., 2015; Liu et al., 2022; Shaloo et al., 2022). To find out the weld surface anomalies, the visual inspection (VI) methods with magnifiers bore scopes or digital cameras are commonly utilized. However, the aforementioned methods strongly depend on operator expertise, sensitivity to ambient noises or lighting, low throughput, and inconsistent weld results in complex, dangerous production environments (Zhang et al., 2022; Gao et al., 2023; Kwon et al., 2023; Lappin et al., 2025). Therefore, many DL approaches such as YOLO (You Only Look Once) V7 (Xu & Li, 2024), VGG16 (Kumaresan et al., 2023; Haohan et al., 2024), MGNet (Yuan et al., 2024), and MobileNetV2 (Ding et al., 2022) have been applied to identify and classify the weld surface defects. And then, these algorithms are integrated in the hardware and industrial robots for automatically detecting weld seams and defects (Soares et al., 2019; Lei et al., 2020; Li et al., 2023; Zhao et al., 2024; Zou et al., 2024; Lappin et al., 2025; Qadoori et al., 2025).

Beyond monocular RGB segmentation, laser-vision pipelines that combine dynamic ROIs with optimized RANSAC have demonstrated real-time seam extraction under arc light and spatter (≈30 fps; millisecond-level processing with early termination and slope constraints) (Chen et al., 2025). In RGB settings, two-stage cascaded designs reduce global computation by first localizing candidate regions and then segmenting only those regions; a representative YOLOv5s–U-Net system reports ≈0.08 s/image end-to-end (Li et al., 2025). Additionally, the improved YOLOv8s-seg models with lightweight backbones and TensorRT on NVIDIA Jetson Nano achieved 97.8% accuracy and took 54 ms from segmentation to path fitting (Zhao et al., 2024).

In RGB settings, two-stage cascaded designs reduce global computation by first localizing candidate regions and then segmenting only those regions; a representative YOLOv5s–U-Net system reports ≈0.08 s/image end-to-end (Li et al., 2025).

Recently, YOLO models and their modified versions have been utilized for the weld-defect localization and segmentation (Redmon et al., 2016; Gao et al., 2023; Kwon et al., 2023). For example, YOLO-Weld, context-aware YOLO, and lightweight LF-YOLO have been applied for weld detection with reduced inference latency and precise weld localization in industrial environment (Liu et al., 2023; Zhang & Ni, 2023; Aboulhosn et al., 2024). In addition, YOLOv8 unifies detection and segmentation in a single network. It allows for effective extraction of pixel-level weld-boundaries without a separate mask-refinement stage (Kwon et al., 2023; Terven & Cordova-Esparza, 2023; Zhao et al., 2024; Zhou et al., 2024). However, converting a segmented boundary into coordinates that a robot can locate is non-trivial. For instance, classic seam-tracking systems rely on handcrafted edge detectors controlled by a proportional controller struggle with occlusion, spatter, and multi-layer beads (Li et al., 2017; Oh et al., 2020; Penekalapati et al., 2022). Moreover, depth-camera and stereo-vision approaches are not straightforward for fitting weld curves (Soares et al., 2019; Lei et al., 2020; Li et al., 2023; Zou & Wan, 2024). Recently, the choice between linear and quadratic (non-linear) regression, and whether to fit the functions X = f(Y) or Y = f(X), is seldom discussed, even though it directly affects weld trajectory stability and accuracy (Zhao et al., 2024). To bridge these gaps, a YOLOv8-based weld-seam segmentation is integrated with a regression-based reconstruction pipeline that automatically selects the dominant axis and applies quadratic fitting.

2. METHOD

Figure 1 presents the proposed pipeline/procedure comprising weld image capture under controlled illumination, YOLOv8-seg for seam segmentation, boundary sampling and centerline extraction, linear/quadratic regressions to obtain a smooth parametric path, pixel-to-world calibration to

{OXYZ}, trajectory smoothing, and execution on the 3-axis robot for the weld seam inspection.

It is worth noting that all experiments were conducted inside a closed inspection cell with dedicated ring illumination; ambient light is blocked by the enclosure, so external environmental variation (daylight, smoke, and airflow) is negligible.

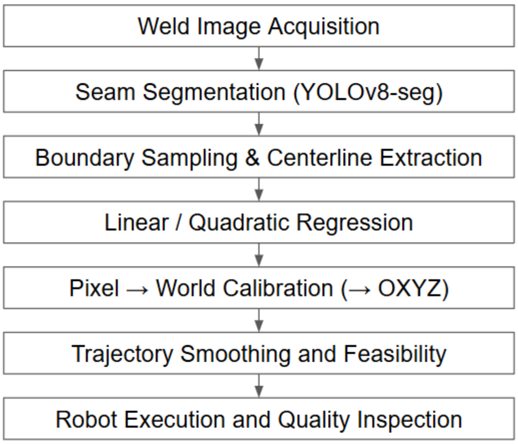

Figure 1. End-to-end workflow from weld image acquisition to robot trajectory

2.1. Experimental setup

Figure 2 presents a three axis robot equipped an IMX219-200 camera with ring illumination, the weld specimen, and an embedded controller (NVIDIA Jetson Nano; Ampere GPU with 1024 CUDA cores and 32 Tensor Cores; 6-core Arm Cortex-A78AE CPU; 8 GB LPDDR5, 128-bit, 102 GB/s; microSD slot; M.2 NVMe Key-M; power modes 7/15/25 W) to set up the experiment for the workflow shown in Figure 1. Weld images are acquired by the Jetson, YOLO-based segmentation and the regression pipelines are executed onboard, and motion commands are streamed to the robot controller. The IMX219-200 (SKU 16679) provides 8 MP (3280 × 2464) on a 1/4″ CMOS (IMX219) sensor with f/2.0, 0.87 mm focal length, ≈200° diagonal FOV, distortion < 11%, four mounting holes, a 15-pin FFC, optional 3.3 V external power, and 25 × 24 mm dimensions. Geometry (camera pose, viewing angle, field-of-view (FOV)) and the data/command flow are detailed in this section and the subsequent method subsections; calibration and pixel-to-world transformation are shown in Section 2.6.3.

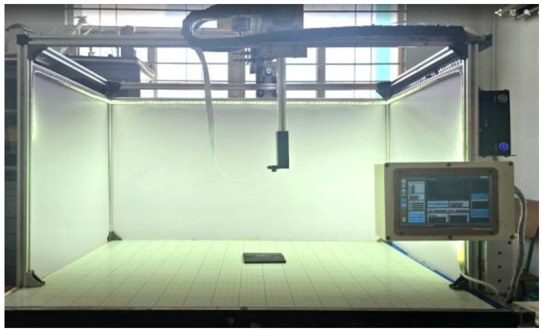

Figure 2. Experimental setup of the robot–vision system (3-axis robot, IMX219-200 camera with ring illumination, weld specimen, and NVIDIA Jetson Nano controller).

YOLO models were trained offline on a separate workstation (Intel® Core™ i7-9700K @ 3.60 GHz, 32 GB RAM, NVIDIA GeForce RTX 2070 SUPER GPU, system-reported graphics memory: 24,332 MB). This workstation with RXT GPU is used only for training/ablation, while all online inference and robot execution run on the Jetson controller with the Ampere GPU.

2.2. Three-axis robot for weld image collection

A three-axis robot is utilized to acquire part of the dataset offline. Weld specimens placed on the robot workspace are imaged by an industrial camera (Basler ace acA3800-14uc, Basler, Germany). Camera's motion is driven by a RC-servo motor with the commands sent from an Arduino microcontroller to ensure adequate coverage of weld contours during the data acquisition. This setup is for data collection only. The weld inspection system is explained in the Section 2.1 (IMX219-200 camera + NVIDIA Jetson Nano). The mechanical design of the three-axis robot is described in detail in the previous work (Tai & Dung, 2024).

The three-axis robot is equipped with three 1.8^0 stepper motors. The timing belt mechanism is designed for the X and Y axes to ensure their smooth and rapid movement, while the ball screw mechanism is built for the Z-axis to enhance precise vertical positioning. The robot's specifications are shown in Table 1. The industrial camera is installed at the robot's end effector for weld dataset collection. The robot is controlled by an Arduino Mega 2560 microcontroller. It receives data from image processing software programmed in Python via a USB interface. Three axes of the robot are controlled using pulse width modulation (PWM) signals to precisely execute pre-computed trajectories. This design achieves a balance between speed, flexibility, and accuracy, making it suitable for industrial weld inspection applications (Nguyen & Nguyen, 2024).

Table 1. Specifications of robot's axes

Parameters	X axis	Y axis	Z axis
Transmission mechanism	Timing belt	Timing belt	Ball screw
Movement range (mm)	0 - 300	0 - 250	0 - 200
Precision (mm/step)	0.2	0.2	0.04

The robot is equipped with an industrial camera to capture weld surface images supporting the detection and segmentation processes. The camera is securely mounted on the Z-axis of the robot via a bracket to ensure stability and allow effective positional adjustments. The optical axis of the camera is aligned perpendicularly to the weld surface to minimize image distortion and ensure accuracy in detection.

The camera with the USB 3.0 serial protocol is directly connected to the computer in which the proposed image processing and segmentation algorithms are implemented. The technical specifications of the camera are described in Table 2. To maintain image quality during operation duration, an industrial LED lighting system is utilized to reduce the effects of external lighting conditions and ensure uniform illumination. An image-processing approach developed in Python performs segmentation and preprocessing to precisely address weld trajectories.

Table 2. Camera specification

Parameter	Value
Sensor Type	CMOS
Resolution	1920 x 1080 pixels
Frame Rate	30 FPS
Interface	USB 3.0
Lens	8 mm focal length

2.3. Data acquisition and preprocessing

The dataset utilized in this study consists of 1,691 1G weld images. They were collected under control illumination and categorized into four groups to reflect the actual welding performance: (1) Non-Acceptable Welds with significant defects such as porosity, misalignment, and surface inconsistencies (named LV1); (2) Welds with common errors and diverse quality (named LV2); (3) Basic welds with good quality (named LV3); (4) Complex welds with higher quality (named LV4). The weld image acquisition method is described in detail elsewhere (Tai & Dung, 2024), The number of images in each group is appropriately allocated to ensure a balanced distribution for model training, validation, and testing.

The data comprise 1G weld images captured by an industrial camera from actual 1G welding processes regardless of lighting conditions, smoke, or reflections. Images are taken from multiple angles to enhance diversity and robustness for the collected dataset. The dataset is then divided into three subsets (Table 3). Training set uses 1,208 images (71.5%) for model training. Validation set comprises 313 images (18.5%) for hyper-parameter tuning and performance monitoring. And 170 images (10%) are utilized for the final evaluation of model performance:

Training set consists of 1,208 images (71.5% of the total dataset). They are utilized to train the model. It includes various weld types (LV_j, j = 1-4), enabling the model to learn important features for weld region detection and segmentation. The objective is to optimize the model weights, ensuring its ability to capture complex characteristics.

Validation set comprises 313 images (18.5% of the total dataset). This set is utilized to evaluate the model's performance during training. It provides critical feedback on hyper-parameters such as learning rate and batch size, helping to adjust the model and prevent over- or under-fitting.

Test set includes 170 images (10% of the total dataset). This test set is used for the performance evaluation. It is able to measure the model's generalization ability by using an entirely new dataset. This ensures accuracy and reliability in detecting weld seams and defects.

The data preprocessing steps include: resizing images to 416×416 pixels to meet the input requirements of YOLOv8, adjusting brightness and contrast to highlight the weld regions, applying noise reduction to eliminate unwanted background elements, and implementing data augmentation techniques such as rotation, flipping, zooming, and scaling to enhance data diversity and mitigate the risk of over fitting during model training (Kwon et al., 2023; Zhang & Ni, 2023; Aboulhosn et al., 2024).

2.4. Weld segmentation and boundary detection

The computer utilized for the training process is as follows: (1) Intel(R) microprocessor with core(TM) i7-9700K and CPU speed of 3.60GHz (8 CPUs); (2) Memory of 32,768 MB RAM; (3) Page file with 14,507 MB used and 23,043 MB available; (4) GPU is NVIDIA GeForce RTX 2070 SUPER with 24,332 MB RAM. Training hardware is described in Section 2.1.

Table 3. Weld image dataset split for training, validating, and testing

Groups	No. Images	Splitting		
		Training	Validating	Testing
LV1	551	408	113	30
LV2	580	400	100	80
LV3	280	200	50	30
LV4	280	200	50	30
Total	1,691	1,208 (71.5%)	313 (18.5%)	170 (10%)

To achieve a balance between accuracy and training efficiency, the model's parameters are described as follows: (1) the number of epochs is set to 100. It is determined through initial experiments. This ensures sufficient time for the model to learn complex data features without over-fitting; (2) batch size is set to 16. This value is optimized based on hardware processing capabilities. A smaller batch size reduces memory requirements and allows frequent weight updates to ensure training stability for certain segmentation tasks; (3) a decay learning rate strategy is utilized. It starts with a high value to enable the model to learn fundamental features quickly. And then it gradually decreases to refine details. This approach prevents the model from getting trapped in local minima and improves overall accuracy; (4) data augmentation consists of rotation, flipping, scaling, brightness adjustment, and contrast enhancement techniques. They are applied to increase dataset diversity. This helps the model to learn features under varying real-world conditions, minimizing over-fitting and improving generalization performance. It is noted that the inference parameters are set 416×416 and FP16. Additionally, model-based FPS is measured on the RTX 2070 SUPER workstation (TensorRT), whereas the pipeline FPS is addressed on Jetson Orin Nano (see Section 2.5).

2.5. Designing a weld seam mask

The algorithm for designing the weld seam mask is described in Figure 3. The weld seam mask is designed based on the raw mask (the weld region filled with a color) predicted by YOLOv8-Seg (Figure 4a). The objective of designing the weld seam mask is to figure out the centerline on which the weld seam trajectory is generated to help the robot move on.

Step 1: The predicted weld region by YOLOv8-seg is utilized to address the outer contour to produce a clear mask on the raw image (see Figure 4b).

Step 2: The found mask from step 1 is uniformly sampled to obtain points lie on the weld boundary. Casting rays along ± the normal within weld seam width (W) is to find opposite edges, then taking midpoints to form set S. Finally, smooth and prune short spurs to obtain continuous centerline points (Figure 4c).

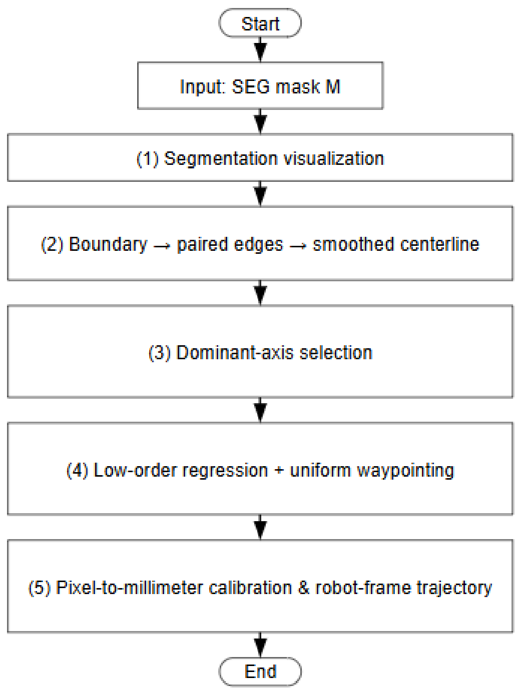

Figure 3. An algorithm for designing the weld seam mask to create the movement robot trajectory

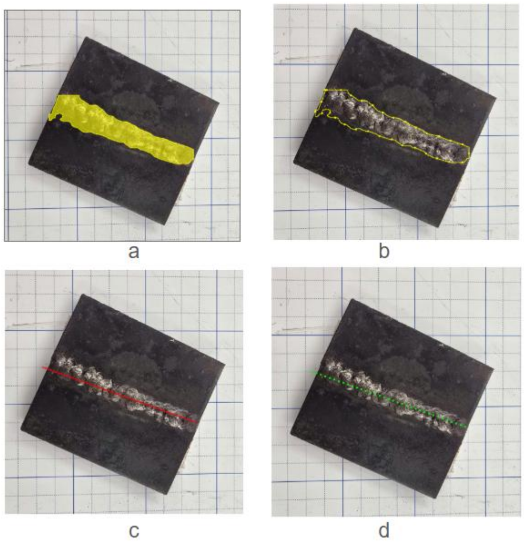

Figure 4. Weld seam masks before (a) and after applying the algorithm of Figure 3 (b, c, and d).

Step 3: Improving the stability by fitting along the axis with larger variance: if $\mathrm{Var}(x) \geq \mathrm{Var}(y)$, $y = f(x)$ is selected, otherwise $x = f(y)$.

Step 4: A linear or quadratic regression model is utilized to estimate the centerline from the smoothed *S* set from Figure 4c and uniformly resample to generate evenly spaced points shown in green dots (Figure 4d).

Step 5: All pixel coordinates of the green dots of the centerline from step 4 are scaled to the {OXYZ} coordinate according to Zhang's method (using standard 10×10 mm squares) (Burger, 2016). This {OXYZ} coordinate in mm could be utilised for the movement of the robot trajectory.

2.6. Nonlinear regression-based weld seam construction

2.6.1. Extraction of feature points

The process of extracting feature points from weld boundaries involves segmentation algorithms, boundary detection, smoothing, and uniform sampling to construct the robot's weld seam trajectory. First, YOLOv8 model is applied to segment objects in the image and generate a binary mask $M(x, y)$ (Sapkota et al., 2024) as follows.

$$M(x,y) = \begin{cases} 1, if\,(x,y) \in weld\ region, \\ 0, otherwise \end{cases} \quad (1)$$

The mask is resized to match the original image dimensions using the OpenCV libraries for the next boundary detection step. Then the findContours algorithm is utilized to detect contours *C* of all boundary points from the binary mask $M(x,y)$ in equation (1). A contour is defined as a series of consecutive boundary points on the mask, represented as follows.

$$C = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \quad (2)$$

where (x, y) refers to the pixel positions of the weld image boundary. n is the number of pixels. The boundary often contains noise or unnecessary points. Therefore, to minimize noise and smooth the boundary, the approxPolyDP algorithm is applied with the step size, Δs. It is worth noting that the larger the Δs is, the fewer the approximate C' boundary points are. This algorithm uses the Douglas-Peucker principle to interpolate and smooth the boundary, with an approximated set of points C' calculated as follows.

$$C' = \{(x_1', y_1'), (x_2', y_2'), \cdots, (x_m', y_m')\} \quad (3)$$

where $(C, C') \leq \varepsilon$. ε is the allowable error threshold, which removes redundant points while preserving the overall shape of the boundary. Finally, the feature points (SEG points, *P*) are extracted from the smoothed boundary C' using a uniform sampling method with a step size/distance Δs. The SEG points are selected by iterating over the boundary points at a fixed distance, as shown below.

$$P = \{(x_1^*, y_1^*), (x_2^*, y_2^*), \cdots, (x_r^*, y_r^*)\} \quad (4)$$

where (x^*, y^*) refers to the boundary points (r) after smoothing and uniform sampling. To address a number of expected boundary points (k), the following form is utilized $k = \mathrm{round}\,(r/\Delta s)$.

The step size or distance is adjusted based on the requirements for accuracy and processing speed. These SEG points are not only stored in an array but also visualized on the image by marking the points and drawing the corresponding boundary. This process ensures that the weld boundary data is optimally extracted and utilized to construct the robot's accurate motion trajectory.

2.6.2. Establishing the regression equation

The objective of this work is to construct a model to accurately describe the weld boundary using a nonlinear regression equation. This estimated equation is utilized for the robot's movement trajectory. Moreover, this model is evaluated based on its accuracy and suitability by using the different types of input weld images such that the residual error (E) is minimized. This error is defined as the sum of the squared deviations between the actual values of the boundary points and the predicted values from the model expressed as follows.

$$E = \sum_{i=1}^{r} \left(y_i^* - f\left(x_i^* \right) \right)^2 \quad (5)$$

where (x_i, y_i) stands for the i^{th} boundary point. $f(x) = a \times x^2 + b \times x + c$ is a quadratic regression model. The coefficients of the model, a, b, and c are determined through the nonlinear regression. The performance of the proposed regression model is evaluated based on the residual sum of squares (RSS) described as follows.

$$RSS = \sum_{i}^{r} \left(y_i^* - f\left(x_i^* \right) \right)^2, \quad (6)$$

and the mean square error $R^2 = 1 - RSS/TSS$. TSS is the total sum of square calculated as follows.

$$TSS = \sum_{i}^{r} \left(y_i^* - \overline{y^*} \right)^2 \quad (7)$$

where $\overline{y^*}$ refers the average value of boundaries respect to y^*.

2.6.3. Constructing the weld seam trajectory

The constructed trajectory from the regression equation is transformed into the robot's coordinates based on the image processing. The image coordinate system is shown in pixels whereas the robot control system requires absolute coordinates in millimeters (mm). This transformation uses scaling factors k_x and k_y, along with the conversion from the relative coordinate system to the absolute coordinate system: $X = x_i^* k_x, Y = y_i^* k_y, Z = z_i^* k_z$.

Based on the regression equation $f(x)$, a set of evenly spaced points along the X-axis (or Y-axis) is computed as follows.

$$Q = \left\{ \left(X_1, Y_1 \right), \left(X_2, Y_2 \right), \cdots, \left(X_r, Y_r \right) \right\}, \quad (8)$$

where $Y_r = f(X_r)$, it is noted that the spacing between points along the x-axis is adjusted to balance the smoothness of the trajectory and processing efficiency.

It is necessary to transform them into the OXYZ absolute coordinate system to precisely control the robot. This process uses the transformation matrix T

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = T . \begin{bmatrix} x_i^* \\ y_i^* \\ z_i^* \\ 1 \end{bmatrix}, \quad (9)$$

where T is a homogeneous matrix representing the affine transformation from the relative coordinate system to the OXYZ absolute coordinate system. The elements of T are determined based on the position and orientation of the camera or robot in space. Figure 5 describes how to transform from the pixel coordinate system to the absolute/three-axis robot coordinate system. Once the absolute coordinates (OXYZ) are defined in the robot's working space, this set of points is utilized to generate control commands.

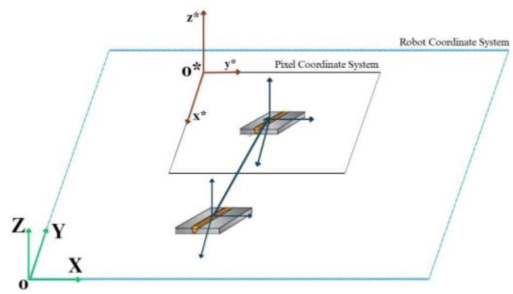

Figure 5. Transformation of absolute space (OXYZ) from the pixel coordinate (Ox^* y^* z^*)

3. RESULTS AND DISCUSSION

3.1. Segmentation and training

The YOLOv8-based segmentation model was utilized to segment and locate weld regions. The experimental results demonstrate that the proposed model gives high accuracy across a diverse dataset. The box mAP50 score reaches 97.5%. It reflects the model's capability to accurately locate weld regions in the images. Simultaneously, the segmentation's mAP50 achieved 98.3%. It shows that the model's reliability for segmenting weld boundaries is high. Moreover, the precision and recall are 97.2% and 97.7%, respectively (see Table 4 and Figure 6). In short, the proposed model not only accurately identifies the target objects but also effectively detects all required regions for segmentation.

Model based offline inference with the GPU of workstation computer (RTX 2070 SUPER, 416×416, FP16, batch=1, TensorRT) reaches 27 FPS, excluding I/O processing and post-processing stages. For the configuration of NVIDIA Jetson Nano with Ampere GPU, the end-to-end pipeline operates at 2.9 FPS. The FPS is computed by counting the number of images captured during the finished processing time. Concurrently, pre/post-processing and regression related to inference yield an average pipeline latency of 346 ms.

Table 4. Segmentation and training performance

Box mAP50	Box mAP50-95	Box Precision	Box Recall	Seg. mAP50	Seg. mAP50-95	Seg. Precision	Seg. Recall
0.976	0.890	0.954	0.962	0.983	0.864	0.972	0.977

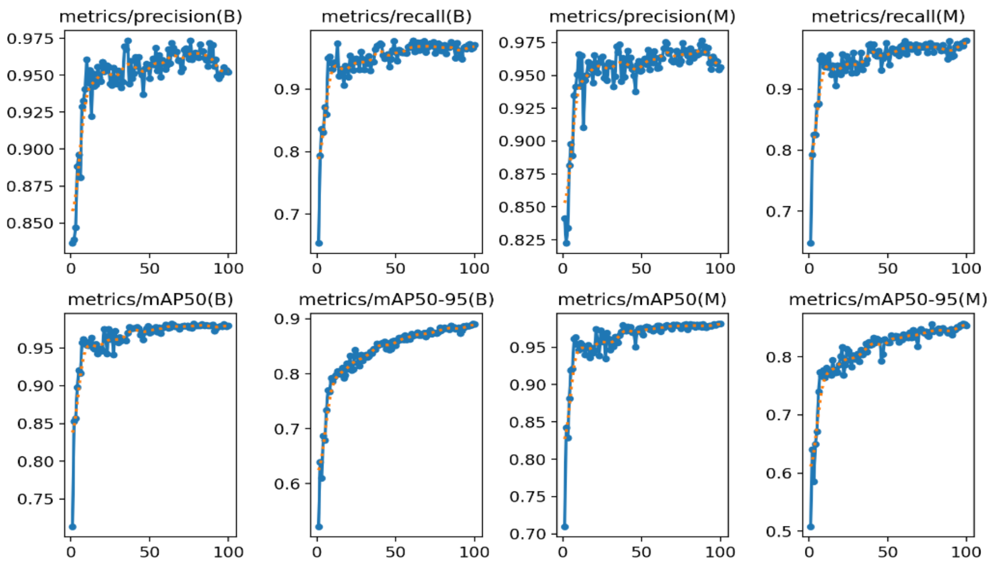

Figure 6. Weld segmentation performance

Figure 7 shows the confusion matrix. True welds are correctly detected at 95%, although background confusion remains; this can be mitigated by more uniform illumination. However, the model still faces the background. This can be overcome by reinforcing the external light for uniform illumination.

The experimental results show that the YOLOv8 based segmentation performance is approximately 98% for different kinds of dataset including welds with significant defects and non-compliant welds (LV1 and LV2) and high-quality welds (LV3 and LV4). This diversity provides a comprehensive assessment of the model's ability to identify and segment welds under various surface conditions. This highlights the model's flexibility and strong adaptability to real-world industrial environments, where input data is often inconsistent and subject to issues such as lighting variability, defects, and background noise. Additionally, the proposed approach is suitable for automated inspection systems.

In short, the trained YOLOv8 model is utilized to perform segmentation and detect the weld boundary regions in input images. The results show that the segmentation mask is clear and accurate (see Figure 8). It correctly describes the weld boundary. Based on this segmentation mask, the findContours algorithm in the OpenCV library is applied to detect and extract the weld boundary (see Section 2.6.1).

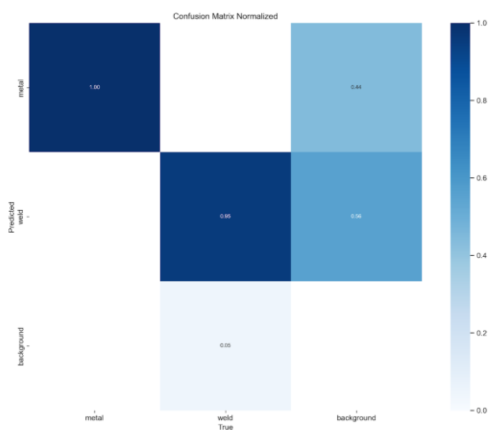

Figure 7. Confusion matrix

A control interface, designed using Tkinter (see Figure 8), provides users with a visual representation of the image-processing results and the weld-boundary extraction process. This interface allows for displaying the segmented image, the detected boundaries, and the feature SEG points marked along the boundary. Additionally, the interface offers a feature to adjust the step size parameter, enabling flexible control over the spacing between SEG points to meet specific requirements for constructing the robot's trajectory.

Figure 8. The segmented image and the detected boundaries based on Tkinter

The results of boundary extraction from the segmentation mask demonstrate that the weld boundaries are accurately identified with uniformly spaced SEG points interpolated and clearly marked along the boundary. Figure 9 presents an example of a weld image with the predicted segmentation mask overlaid (yellow curve) compared with the expert SEG (magenta curve). The points of the weld seam boundaries are sampled along with contours to figure out the mid-points. The centerline (red line) is addressed by the quadratic regression method of the mid-points computed from paired edges within the mask. To reconstruct the weld seam trajectory for the robot movement, the centerline is uniformly resampled to generate evenly spaced points. The experimental results illustrate that the system operates reliably, correctly detecting the weld boundary regions while allowing for flexible adjustment of the SEG point density as needed.

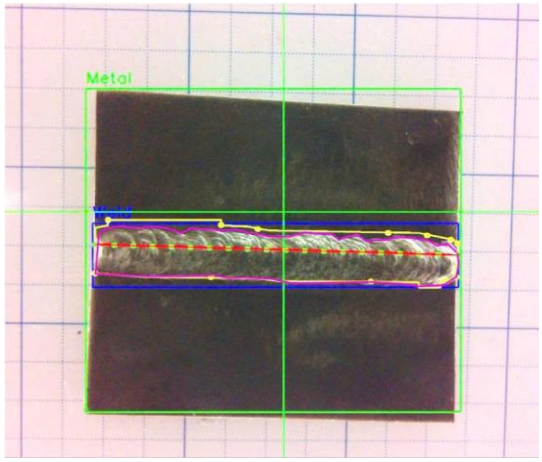

Figure 9. Boundary extraction from the segmentation mask

3.2. Regression models

Figure 10 presents the fitting lines estimated from the linear ($f(x)$ = a×x + b) and nonlinear ($f(x) = a \times x^2 + b \times x + c$) regression models. It is noted that both models can perfectly describe the boundary points. The linear regression shows a good fit to the data, particularly when analyzed through the equations $Y = f(X)$ and $X = f(Y)$. Both linear equations achieve an $R^2 = 0.94$. It means they can describe the weld seam. However, the *MSE* and *RSS* of the equation $Y = f(X)$ are lower than those of $X = f(Y)$ (Table 5). This may be related to the characteristics of the data distribution that allow reduce noises during the regression process. The quadratic (non-linear) regression model is similar to the linear approach ($R^2 = 0.94$). Both nonlinear models of $X = f(Y)$ and $Y = f(X)$ have *RSS* and *MSE* that are slightly smaller than those of the linear models. These findings imply that if the weld-boundary dataset exhibits an inherently linear trend, both linear and nonlinear models are suitable for fitting the weld boundaries.

Table 5. Performance of regression models with 48 boundary SEG points

Model	Estimated equations	MSE	RSS	TSS	R²
Linear Regression	$X = -1.28Y + 3{,}747.98$	8.34	400.13	6,165.88	0.94
Linear Regression	$Y = -0.73X + 2{,}812.04$	4.73	227.02	3,498.29	0.94
Polynomial Regression	$X = -0.0001Y^2 - 1.05Y + 3{,}609.24$	8.30	398.30	6,165.88	0.94
Polynomial Regression	$Y = -0.0000X^2 - 0.55X + 2{,}632.62$	4.71	225.95	3,498.29	0.94

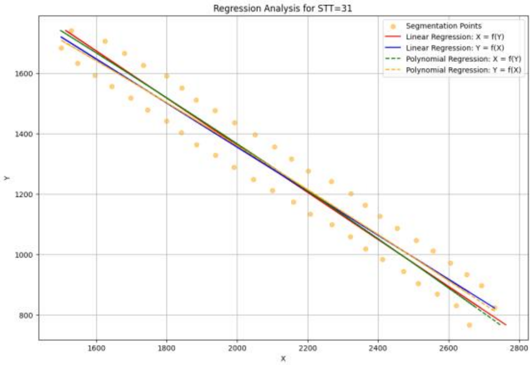

Figure 10. Regression models with 48 boundary SEG points

Subsequently, using the same weld image data shown in Figure 10 and Table 5, but doubling the step size to reduce the number of SEG points from 48 to 24, the results are presented in Figure 11 and Table 6.

The experimental results show that the performance of the regression models on MSE and R^2 is similar to that in the case of 48 SEG points. However, the RSS and TSS are reduced by half compared to those in the case of 48 SEG. This demonstrates that 24 SEG points are still sufficient to accurately describe the weld boundary, although there are minor changes in the performance evaluation metrics. It means even though a number of SEG points are reduced, the regression models are still ensured for both MSE and R^2. However, they are reduced to 16 SEG points, and the performance of the regression models is reduced. The R2 value is approximately reduced a half, 0.52 (Table 7).

Table 6. Performance of regression models with 24 boundary SEG points

Model	Estimated equations	MSE	RSS	TSS	R²
Linear Regression	$X=-1.27Y + 3{,}731.81$	8.43	202.20	3,066.84	0.93
Linear Regression	$Y= - 0.73X + 2{,}824.50$	4.87	116.80	1,771.46	0.93
Polynomial Regression	$X= - 0.0001Y^2 - 0.94Y + 3{,}529.69$	8.34	200.11	3,066.84	0.93
Polynomial Regression	$Y = - 0.0000X^2 - 0.55X + 2{,}632.40$	4.84	116.20	1,771.46	0.93

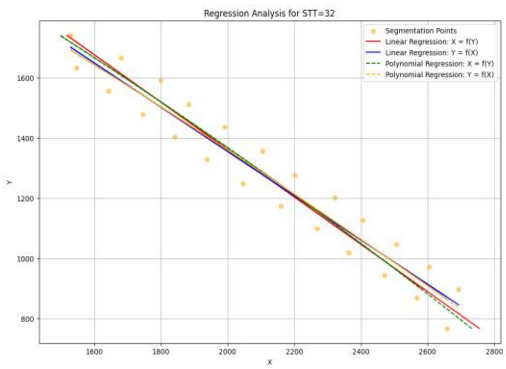

Figure 11. Regression models with 24 boundary SEG points

Overall, the two regression paradigms differ only marginally for *MSE* and R^2. The non-linear fit offers a slight advantage in goodness of fit but demands denser SEG boundary points to preserve that benefit, whereas the linear formulation, particularly Y = f(X), remains performance with the reduction of SEG boundary points. Additionally, the linear regression is economical and adequate for straight or quasi-straight seams but incurs larger RSS/MSE when curvature or noise is present, whereas quadratic regression accommodates smooth, continuous curves with slightly higher fidelity. Of course, the cost of processing requirements for denser weld boundary points is also higher.

In short, the experimental results demonstrated that 24 SEG points give a similar performance as 48 SEG points, while it is poor in the case of 16 SEG points. Therefore, the 24 SEG points are recommended to be utilized in the current work.

Table 7. Performance of regression models with 16 boundary SEG points

Model	Estimated equation	MSE	RSS	TSS	R²
Linear Regression	$X = 0.34Y + 1{,}342.25$	8.25	148.47	309.13	0.52
Linear Regression	$Y = 1.51X - 1{,}300.86$	36.37	654.61	1,363.00	0.52
Polynomial Regression	$X = 0.0001Y^2 + 0.25Y + 1{,}406.48$	8.24	148.38	309.13	0.52
Polynomial Regression	$Y = - 0.0005X^2 + 3.28X - 2{,}944.50$	36.29	653.24	1,363.00	0.52

3.3. Weld seam trajectory

The resulting centerline is transformed from image space to a metric (millimeter) trajectory and expressed in the robot frame {OXYZ} (see Section 2.6.3 and Figure 5). Experiments were conducted on 1,691 images labeled 1G. Under the deployed pipeline configuration on NVIDIA Jetson Nano, the system operates at 2.9 FPS. Under the closed-cell setup (shown in Section 2.1), repeated runs with identical camera settings produced consistent trajectory accuracy and throughput. The mean absolute trajectory error is 0.24 mm, computed by comparing the estimated weld seam trajectory and the labeled actual weld seam centerline (expert SEG).

Once the weld-seam trajectory is reconstructed from the segmented boundaries, the three-axis robot can be used to inspect weld defects. In addition, modern control algorithms, such as the reference-model adaptive controller, can be applied to precisely locate the actual weld positions (Qadoori et al., 2025).

3.4. FPS and time response of the proposed deep-learning model

To evaluate the efficiency of the predicted raw mask produced by the Yolov8-seg model, the response time and latency are compared with those of two deep learning models, U-Net and DeepLabv3+. It is noted that these computations do not include estimating the midpoints on the centerline (performed on a workstation) and converting from the centerline to the robot's motion trajectory (performed on the Nano Jetson). The input for all three models used the same 217 labeled weld images. Other parameters, such as imgsz = 640, FP16, and batch = 1, are set identically for all three deep learning models aforementioned. And a higher FPS value is better whereas a lower latency (measured in ms/image) is better. Table 8 presents comparative results for processing speed and latency across three deep learning models (YOLOv8-Seg, DeepLabv3+, and U-Net). The experimental results showed that the latency of YOLOv8-seg is the lowest (33.5 ms/image) compared with DeepLabv3+ (165.0 ms/image) and U-Net (59.1 ms/image). Additionally, among the three deep learning models aforementioned, the processing speed (FPS) of YOLOv8-Seg is the best (29.9), while DeepLabv3+ and U-Net are 6.1 and 16.9, respectively. It is for this reason that YOLOv8-seg is utilized in the current study to predict the raw masks.

Table 8. Comparison of the deep-learning models performance (FPS and time response)

Models	FPS	Latency (ms/image)
DeepLabv3+	6.1	165.0
YOLOv8-Seg	**29.9**	**33.5**
U-Net	16.9	59.1

4. CONCLUSION

This study presented a method for extracting and optimizing the weld seam trajectory using a YOLOv8 segmentation model and regression algorithm. Then the trajectory was transformed from the weld image space into the OXYZ coordinate system to generate commands for moving the weld inspection robot. The experimental results showed that both methods give similar R^2 and *MSE*, while RSS and TSS of the nonlinear regression model are slightly smaller than those of the linear model. Additionally, if a number of boundary SEG points were reduced by half, the values of *RSS* and *TSS* were also decreased accordingly. In future works, the proposed approach will be deployed in the automated weld inspection application for practical automated inspection by using a robot-equipped camera and high-speed image processing software.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Aboulhosn, Z., Musamih, A., Salah, K., Jayaraman, R., Omar, M., & Aung, Z. (2024). detection Detection of manufacturing defects in steel using deep learning with explainable artificial intelligence. *IEEE Access*, *12*, 99240–99257. https://doi.org/10.1109/ACCESS.2024.3430113

Burger, W. (2016). *zhang's* Zhang's camera calibration algorithm: in-depth tutorial and implementation. *Technical Report HGB16-05.* University of Applied Sciences Upper Austria. https://doi.org/10.13140/RG.2.1.1166.1688/1

Chen, G., Zhang, Y., Ai, Y., Yu, B., & Xu, W. (2025). Real-time seam extraction using laser vision sensing: hybrid approach with dynamic ROI and optimized RANSAC. *Sensors*, *25*(11), 3268. https://doi.org/10.3390/s25113268

Deepak, J. R., Bupesh Raja, V. K., Srikanth, D., Surendran, H., & Nickolas, M. M. (2021). Non-destructive testing (NDT) techniques for low carbon steel welded joints: A review and experimental study. *Materials Today: Proceedings*, *44*, 3732–3737. https://doi.org/10.1016/j.matpr.2020.11.578

Ding, K., Niu, Z., Hui, J., Zhou, X., & Chan, F.T.S. (2022). A weld surface defect recognition method based on improved MobileNetV2 algorithm. *Mathematics,* 10, 3678. https:// doi.org/10.3390/math10193678

Gao, A., Fan, Z., Li, A., Le, Q., Wu, D., & Du, F. (2023). YOLO-Weld: A modified YOLOv5-based weld feature detection network for extreme weld noise. *Sensors*, *23*(12), 5640. https://doi.org/10.3390/s23125640

Hashim, M. H. M., Arifin, A. M. B. T., AzmirOsman, S., Osman, S. A., Kiat, T. S., & Salihin. A. F. M. (2024). Study on MIG and TIG weldment behaviour using finite element approach. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *46*(2), 81–89. https://doi.org/10.37934/araset.46.2.8189

Jolly, M., Prabhakar, A., Sturzu, B., Hollstein, K., Singh, R., Thomas, S., Foote, P., & Shaw, A. (2015). Review of non-destructive testing (NDT) techniques and their applicability to thick walled composites. *Procedia CIRP*, *38*, 129–136. https://doi.org/10.1016/j.procir.2015.07.043

Kumaresan, S., Aultrin, K. S. J., Kumar, S. S., & Dev Anand, M. (2023). Deep learning-based weld defect classification using VGG16 transfer learning adaptive fine-tuning. *International Journal on Interactive Design and ManufacturingInt J Interact Des Manuf*, *17*, 2999–3010. https://doi.org/10.1007/s12008-023-01327-3

Kwon, J. E., Park, J. H., Kim, J. H., Lee, Y. H., & Cho, S. I. (2023). Context and scale-aware YOLO for welding defect detection. *NDT & E International*, *139*, 102919. https://doi.org/10.1016/j.ndteint.2023.102919

Lappin, E., Oubre, J., Gurau V., & Taheri, H. (2025). Automated Robotic and AI-Driven Nondestructive Inspection for Enhanced Welding Flaw Detection, *The 6th International Conference on Artificial Intelligence, Robotics and Control (AIRC),* Savannah, GA, USA, (pp. 99-106). https://doi.org/10.1109/AIRC64931.2025.11077497.

Lei, T., Rong, Y., Wang, H., Huang, Y., & Li, M. (2020). A review of vision-aided robotic welding. *Computers in Industry*, *123*, 103326. https://doi.org/10.1016/j.compind.2020.103326

Li, A., Li, D., & Wang, A. (2025). A two-stage YOLOv5s–U-Net framework for defect localization and segmentation in overhead transmission lines. *Sensors*, *25*(9), 2903. https://doi.org/10.3390/s25092903

Li, M., Huang, J., Xue, L., & Zhang, R. (2023). *A* guidance system for robotic welding based on an improved YOLOV5 algorithm with a realsense depth camera. *Research Square Platform LLC*. https://doi.org/10.21203/rs.3.rs-3256246/v1

Li, X., Li, X., Ge, S. S., Khyam, M. O., & Luo, C. (2017). Automatic welding seam tracking and identification. *IEEE Transactions on Industrial Electronics*, *64*(9), 7261–7271. https://doi.org/10.1109/TIE.2017.2694399

Liu, M., Chen, Y., Xie, J., He, L., & Zhang, Y. (2023). LF-YOLO: A lighter and faster yolo for weld defect detection of X-Ray image. *IEEE Sensors Journal*, *23*(7), 7430–7439. https://doi.org/10.1109/JSEN.2023.3247006

Liu, Y., Yuan, K., Li, T., Li, S., & Ren, Y. (2022). NDT method for line laser welding based on deep learning and one-dimensional time-series data. *Applied Sciences*, *12*(15), 7837. https://doi.org/10.3390/app12157837

Mirapeix, J., García-Allende, P. B., Cobo, A., Conde, O. M., & López-Higuera, J. M. (2007). Real-time arc-welding defect detection and classification with principal component analysis and artificial neural networks. *NDT & E International*, *40*(4), 315-323. https://doi.org/10.1016/j.ndteint.2006.12.001

Nguyen D. T. & Nguyen H. -D. (2024). Application of a 3-axis robot arm combined with a camera supports automatic weld data collection. *TNU Journal of Science and Technology*, *229*(14), 43 - 51. https://doi.org/10.34238/tnu-jst.10782

Nguyen, H.-D., Le, H.-D., Nguyen, V. K., & Lam, H. M. (2024). Development of the intelligent traffic light system based on image processing and fuzzy control techniques. *CTU Journal of Innovation and Sustainable Development*, *16*(3), 9-20. https://doi.org/10.22144/ctujoisd.2024.302

Oh, S., Jung, M., Lim, C., & Shin, S. (2020). Automatic detection of welding defects using faster R-CNN. *Applied Sciences*, *10*(23), 8629. https://doi.org/10.3390/app10238629

Penekalapati, S. V., Ramana, E. V., & Kumar, N. K. (2022). Automatic detection of sub-surface weld defects using machine learning approach. *2022 International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems (ICMACC)* (pp. 412–415). https://doi.org/10.1109/ICMACC54824.2022.10093625

Qadoori, B. , Alaiwi, Y., Mahmoud, S. , Jundi, A., & Mulki, H. (2025). Trajectory tracking control of KUKA KR 6 R900-2 robotic arm for welding applications. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *64*(1), 140–152. https://doi.org/10.37934/araset.64.1.140152

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788). https://doi.org/10.1109/CVPR.2016.91

Sapkota, R., Ahmed, D., & Karkee, M. (2024). Comparing YOLOv8 and mask R-CNN for instance segmentation in complex orchard environments. *Artificial Intelligence in Agriculture*, *13*, 84–99. https://doi.org/10.1016/j.aiia.2024.07.001

Sarker, I. H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, *2*, 420. https://doi.org/10.1007/s42979-021-00815-1

Shaloo, M., Schnall, M., Klein, T., Huber, N., & Reitinger, B. (2022). A review of non-destructive testing (NDT) techniques for defect detection: application to fusion welding and future wire arc additive manufacturing processes. *Materials*, *15*(10), 3697. https://doi.org/10.3390/ma15103697

Soares, L. B., Weis, A. A., Rodrigues, R. N., & Da C. Botelho, S. S. (2019). A robotic passive vision system for texture analysis in weld beads. *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)* (pp. 535–540). https://doi.org/10.1109/INDIN41052.2019.8972289

Sun, H., Xia, L., Zhou, Y., Qian, W., Wang, W., & Zhang, K. (2024). Online detection and evaluation of weld surface defects based on lightweight network VGG16-UNet and laser scanning. *Journal of Manufacturing Processes*, *129*, 292-306, https://doi.org/10.1016/j.jmapro.2024.08.037.

Terven, J., & Cordova-Esparza, D. -M. (2023). A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *Machine Learning and Knowledge Extraction*, *5*, 1680-1716. https://doi.org/10.3390/make5040083

Xu, X., & Li, X. (2024). Research on surface defect detection algorithm of pipeline weld based on YOLOv7. *Scientific Reports*, *14*, 1881. https://doi.org/10.1038/s41598-024-52451-3

Yuan, L., Chen, Y., Tang, H., Gao R., & Wu, W. (2024). A lightweight deep-learning algorithm for welding defect detection in new energy vehicle battery current collectors. *IEEE Sensors Journal*, *24*(13), 21655-21668, doi: 10.1109/JSEN.2024.3398769

Zhang, Y., & Ni, Q. (2023). A novel weld-seam defect detection algorithm based on the S-YOLO model. *Axioms*, *12*(7), 7. https://doi.org/10.3390/axioms12070697

Zhang, Y., Guo, Z., Wu, J., Tian, Y., Tang, H., & Guo, X. (2022). Real-time vehicle detection based on improved YOLO v5. *Sustainability*, *14*(19), 12274. https://doi.org/10.3390/su141912274

Zhao, M., Liu, X., Wang, K., Liu, Z., Dong, Q., Wang, P., & Su, Y. (2024). Welding seam tracking and inspection robot based on improved YOLOv8s-Seg model. *Sensors*, *24*(14), 4690. https://doi.org/10.3390/s24144690

Zhao, M., Liu, X., Wang, K., Liu, Z., Dong, Q., Wang, P., & Su, Y. (2024). Welding seam tracking and inspection robot based on improved YOLOv8s-Seg model. *Sensors*, *24*(14), 4690. https://doi.org/10.3390/s24144690

Zhou, J., Zhang, W., Sui, X., & Chen, Y. (2024). Welding defect detection based on YOLOv8. *Journal of Physics: Conference Series*, *2816*(1), 012045. https://doi.org/10.1088/1742-6596/2816/1/012045

Zou, Y., & Wan, Y. (2024). Robotic seam tracking system combining lightweight segmentation network design and ADMM-based structured pruning. *IEEE Transactions on Instrumentation and Measurement*, *73*, 1–15. https://doi.org/10.1109/TIM.2024.3381666