Preprints.org

Article

Not peer-reviewed version

Research on Worker Assembly Procedure Inspection Based on Improved YOLO and Hand Trajectory

Shaoyi Zhou , Xiancheng Wang * , Zewen Liu , Yifan Jiang , Lang Huang

Posted Date: 15 November 2024

doi: 10.20944/preprints202411.1140.v1

Keywords: assembly inspection; triplet; multi-feature fusion; YOLO; object detection

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Research on Worker Assembly Procedure Inspection Based on Improved YOLO and Hand Trajectory

Shaoyi Zhou, Xiancheng Wang *, Zewen Liu, Yifan Jiang and Lang Huang

College of Science and Technology, Ningbo University, Ningbo 315211, China
* Correspondence: wangxiancheng@nbu.edu.cn

Abstract: Visual inspection detects manual assembly actions to improve quality. Existing methods mainly analyze behaviors in single assembly scenarios using action recognition algorithms, but they cannot adapt to dynamic product switching and varying station environments in manual assembly lines. This paper proposes an adaptive assembly inspection method based on hand motion trajectories. By constructing an assembly inspection model based on the "Component-Tool-Product" triplet, the YOLOv9 model is employed to capture the spatiotemporal hand trajectory data stream. The assembly process is divided into a sequence of transitions between different assembly regions based on the hand's trajectory. The system analyzes the real-time hand trajectories using Dynamic Time Warping (DTW) and dwell time algorithms to obtain real-time assembly flow, allowing for the detection of missed and misassembled components. Additionally, the frame difference method is introduced to extract hand motion features, and an Attention Feature Fusion (AFF) module is used to integrate multi-scale features with motion texture information, enhancing the performance of the object detection model. Experimental results show that the proposed algorithm effectively reduces the false detection rate, with an average detection accuracy of 96% for missed and misassembled behaviors. The improved YOLOv9 model achieves an mAP@0.5 of 93.94%.

Keywords: assembly inspection; triplet; multi-feature fusion; YOLO; object detection

1. Introduction

In specific manufacturing sectors, particularly in the small appliance and small furniture industries, manual assembly remains predominant due to the complex components and numerous flexible connectors. The full implementation of automation still faces significant challenges[1]. These industries typically feature characteristics such as small batch production, high customization, and low profit margins, and continue to rely heavily on manual assembly during the production process, making complete automation difficult to achieve[2]. This production model has several features: First, production lines require a high degree of flexibility, with frequent product model changes, requiring operators to quickly adapt to different assembly processes. Second, part selection and installation heavily rely on manual operations. Workers often perform repetitive tasks under high intensity and fast pace, leading to fatigue, which results in misassembly and missing components, thereby affecting product performance and safety. This increases rework costs and even leads to safety accidents that endanger life and property. Thus, the skill level of operators, adherence to proper procedures, and accurate installation of parts become key factors influencing product quality[3]. However, traditional manual quality control methods are no longer sufficient to meet the demands of modern production. Therefore, developing an intelligent quality control system for manual assembly lines is particularly important. This system should possess characteristics such as adaptability, real-time processing, and non-invasiveness.

Currently, assembly inspection in manual assembly lines primarily depends on manual inspection. However, with the development of machine vision technology, research that leverages its advantages in image processing and pattern recognition is increasing, aiming to address quality issues, reduce human errors, and enhance mistake-proofing capabilities[4]. Traditional vision-based approaches[5–7] typically extract contour features of machine parts using image preprocessing

© 2024 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

techniques (such as filtering, edge detection, threshold segmentation, etc.), and then assess whether the parts have defects through feature matching and analysis. The same approach can be applied to missing part detection by evaluating the presence and correct positioning of components to assess assembly quality[8,9]. However, these methods often face challenges when dealing with irregular or flexible parts and require high demands on the shooting environment. Therefore, monitoring and inspecting the entire assembly process to ensure quality from the source is a more effective solution[10].

Wang Cheng et al. designed a data processing model based on spatiotemporal features using information such as tools and hand keypoints in workshop assembly images, predicting process categories and improving the EfficientNetV2 image classification network for workshop task sequence recognition, achieving effective process flow identification[11]. Yang Y., Wang et al. combined lightweight OpenPose with a self-attention model to integrate skeleton and workpiece features, realizing assembly process detection[12]. Daxin Liu et al. proposed a multi-scale multi-stream graph convolutional network (2MSGCN) for assembly action recognition, capturing operator skeleton data using multi-view RGBD cameras, and optimizing with feature fusion and Ghost modules to improve the accuracy and real-time performance of assembly action recognition[13]. Md. Al-Amin proposed a personalized convolutional neural network system based on human skeletal data, improving model adaptability through transfer learning and iterative enhancement, significantly increasing action recognition accuracy using classifier fusion methods (WASC)[14]. Julian Koch introduced an action recognition method based on Methods-Time-Measurement (MTM), detecting general action elements through skeleton data and using search algorithms to estimate assembly progress, extending action recognition in variant assembly processes[15]. However, these methods fall short when addressing the flexibility requirements of manual assembly production lines. When production schemes and objects change, these models need to be retrained and debugged for different setups, resulting in high training costs.

In summary, current research and applications both domestically and internationally remain focused on single-scenario setups, which fail to meet the needs of complex and dynamic manual assembly lines. Therefore, this paper abstracts worker assembly behaviors into high-dimensional representations, using hand trajectories to describe transitions between different objects in the production process (e.g., components, tools, and assembly subjects). This ensures the proposed inspection method has high generality across different assembly workstations, objects, and steps, enabling effective monitoring of the assembly process through coarse-grained surveillance.

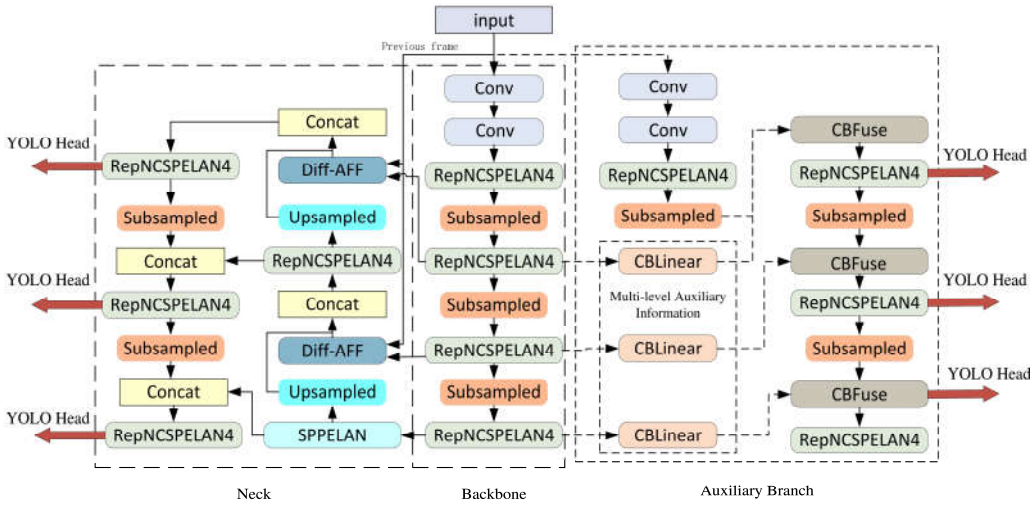
The accuracy of assembly inspection in the above schemes mainly depends on the trajectory accuracy obtained through object detection algorithms. With the rapid development of computer vision technology, various efficient object detection algorithms have been developed and applied. Representative algorithms include Faster-RCNN[16], YOLO, and SSD[17], among which the YOLO (You Only Look Once) algorithm is known for its fast processing speed[18]. YOLO is an object detection system that makes predictions based on global image information. Since the release of its first version in 2015, YOLO has undergone several updates, with each version offering enhanced performance[19]. YOLOv9 is one of the latest object detection models in the YOLO series[20]. Compared to earlier versions, YOLOv9 introduces the concept of Programmable Gradient Information (PGI), generating reliable gradients through auxiliary reversible branches, which helps retain crucial deep features and prevents semantic loss caused by traditional multi-path feature integration[21].

For the YOLOv9 model, S. Yang et al. enhanced small object detection accuracy by adding small-object detection heads, replacing Conv modules with DWConv, and using C3Ghost to replace the backbone module, achieving lightweight deployment[22]. R. An et al. improved the YOLOv9 model by introducing Ghost convolutions, enhancing perception ability and detection accuracy, and deploying the improved model in an intelligent city framework for real-time traffic monitoring[23]. Chun-Tse Chien applied YOLOv9 to fracture detection tasks, improving model performance by training on the GRAZPEDWRI-DX dataset and using data augmentation techniques[24]. Yongxin Chen et al. introduced an efficient multi-scale attention mechanism (EMA) for cross-spatial learning

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

and an improved Inner-SIoU bounding box regression loss function, significantly enhancing defect detection accuracy and convergence speed in substation equipment[25]. Jialin Zou et al. proposed an improved steel defect detection model, CK-NET, which optimizes feature extraction modules, incorporates deformable convolutions and self-attention mechanisms, and improves CBAM and PGI branches to significantly enhance feature extraction and fusion capabilities[26].

This paper focuses on the fixed camera and static background characteristics in assembly inspection. To improve the accuracy of hand recognition in complex environments, this study improves the YOLOv9 model primarily in terms of computational accuracy: 1) Using frame difference to extract spatial features of hand movements, generating differential images that are combined with RGB images from the assembly process, forming a dual-channel input for the object detection network; 2) Introducing an Attention Feature Fusion (AFF) module[27] to integrate spatial hand motion features with high-level semantic features in the feature pyramid. These improvements enhance the model's detection accuracy in complex assembly environments. The final network structure is shown in Figure 1:



Note: Conv is a convolution operation, Subsampled is a downsampling operation, SPPELAN is a spatial pyramid pooling structure, Upsample is an upsampling operation, Diff-AFF is a differential texture feature fusion module, RepNCSPELAN4 is a generalized efficient layer aggregation network, and its combined CBlinear, CBFuse, and Contact modules implement multi-level auxiliary information aggregation of all target object gradient information received by each feature pyramid of the auxiliary reversible branch and pass it to the main branch for weight update.

Figure 1. Improved YOLOv9 model structure

2. Materials and Methods

2.1. Assembly Inspection Method Based on Hand Trajectory

To address the high-cost model training and debugging issues caused by frequent changes in assembly processes, this study proposes an assembly behavior detection method based on hand trajectories and changes in working regions. As shown in Figure 2, to cope with new assembly processes, we propose an assembly behavior detection model based on the "Component-Tool-Product" triplet, aiming to reduce the complexity of modeling new assembly processes. First, workers divide the entire area into several semantic regions based on the structural features of the assembly environment, including assembly zones, part zones, finished product zones, and tool zones, and store the spatial location information of these regions. This division allows the worker's entire assembly behavior to be pre-defined as hand trajectories and transition sequences between different regions, effectively abstracting the essential characteristics of the assembly actions. Additionally, by imposing

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

constraints on the sequence of regional transitions and the dwell time in each region, assembly process detection can be achieved.

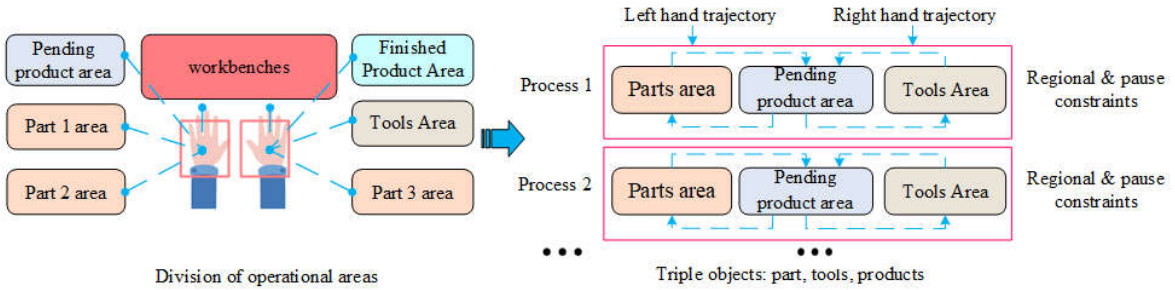


Figure 2. The assembly behavior definition method of the "Component Tool Product" triplet.

The complete hand-trajectory-based assembly detection scheme is shown in Figure 3. The scheme is divided into a pre-learning stage and a real-time detection stage. In the pre-learning stage, the operation space is first divided into different regions, and the spatial domain information of the operation space is obtained: $\mathcal{R} = \{R_1, R_2, \ldots, R_N\}$. Then, several experienced workers demonstrate the standard assembly process. The system uses YOLOv9 to capture the spatiotemporal hand trajectory data stream of skilled workers during assembly, precisely locating their bounding boxes. By combining the saved location information of the work areas, the system can accurately determine the hand's region and obtain the standard process region transition sequence: $S = [R_{s1}, R_{s2}, \ldots, R_{sM}]$. Simultaneously, during each region transition, the system records the operation trajectory within each region, $\mathbf{x}_s(t) = [x_s(t),\ y_s(t)]$, to build a standard process trajectory database.

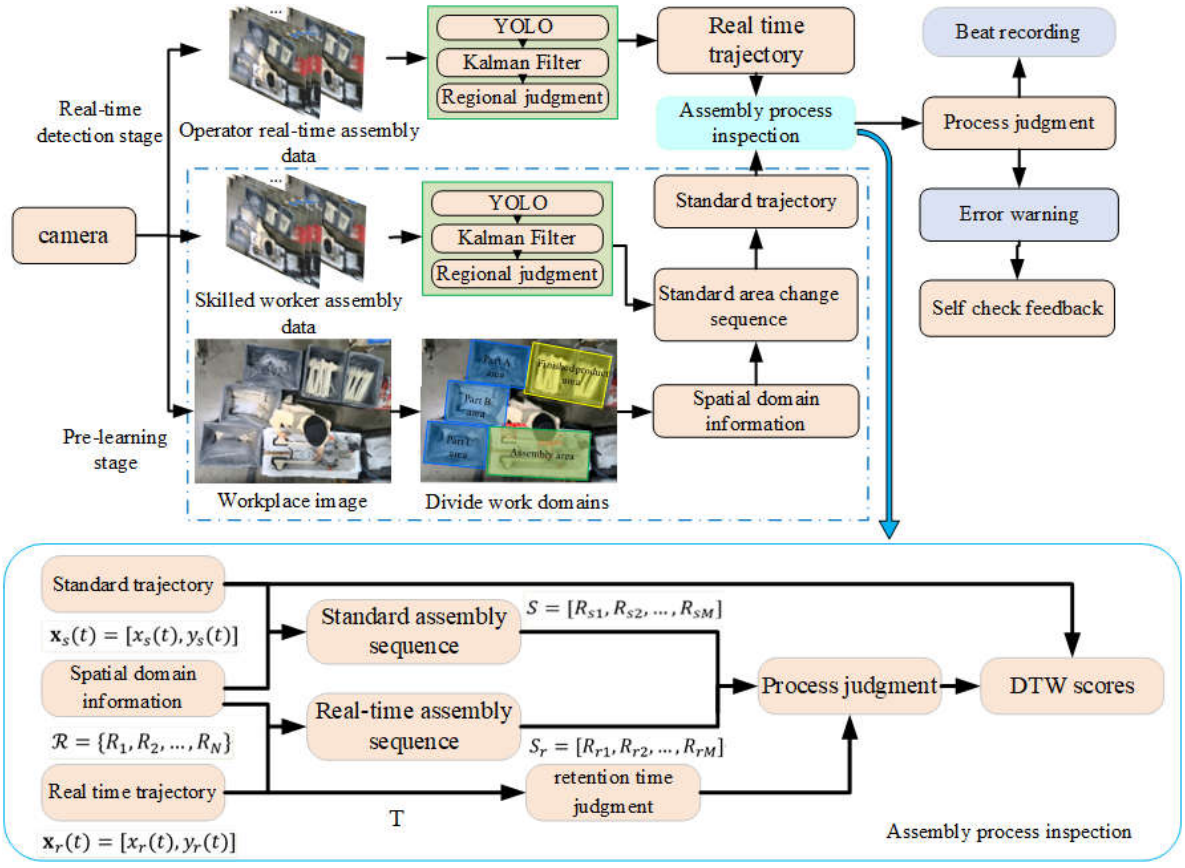


Figure 3. Assembly inspection scheme and framework diagram based on hand trajectory.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

In the real-time assembly detection stage, as the worker operates, the system continuously tracks the hand's trajectory and determines the region. The real-time obtained region transition sequence, $S_r = [R_{r1}, R_{r2}, \ldots, R_{rM}]$, is compared with the preset standard process. When the system detects that the hand's region transition sequence deviates from the standard assembly process (e.g., due to incorrect or missed operations), it alerts the worker via a signal light or other warning forms for self-check. After the worker performs the self-check, the system feedback must be used to confirm the result, ensuring the integrity of the process and the accuracy of the operation, further reducing potential assembly defects.

In tracking the hand trajectory and determining the region, the system employs two main methods to prevent misjudgment. First, it uses the dwell time of the hand in a specified region to make judgments, ensuring that only operations in which the hand stays in the region for a sufficient amount of time are considered valid. Second, by monitoring changes in the hand's acceleration, the system determines whether the trajectory pauses in a certain region. For example, if a worker's hand briefly enters the part zone without performing an actual operation, this strategy effectively prevents misjudgment.

In addition to the judgments based on the region transition sequence and region dwell time, the system also applies the Dynamic Time Warping (DTW) algorithm in real-time to compare the captured real-time hand trajectory, $\mathbf{x}_r(t) = [x_r(t), y_r(t)]$, with the standard process trajectory database. This comparison generates multiple scores, and by averaging these scores, a comprehensive score is obtained, which measures the worker's completion of actions in the current process. This score not only evaluates the worker's assembly accuracy but also further reduces the likelihood of misjudgment, improving the robustness and precision of the system.

Dynamic Time Warping (DTW) is a method used to measure the similarity between two time series of different lengths[28]. Its primary function is to flexibly align trajectory points to assess the similarity between two trajectories. In time series similarity measurement, simple point-to-point similarity calculations are highly susceptible to shifts or misalignments in the sequences. The use of DTW can prevent this issue. DTW employs dynamic programming to align and match two sequences, even if they differ in length, thereby providing a similarity score for the trajectories. As shown in Figure 4, two time series, $A = \{a_1, a_2, \cdots, a_{32}\}$and $B = \{b_1, b_2, \cdots, b_{27}\}$, of different lengths are matched, where the dashed lines between the series indicate the similar points. The DTW algorithm calculates the similarity between the two time series based on the sum of the distances between these similar points.

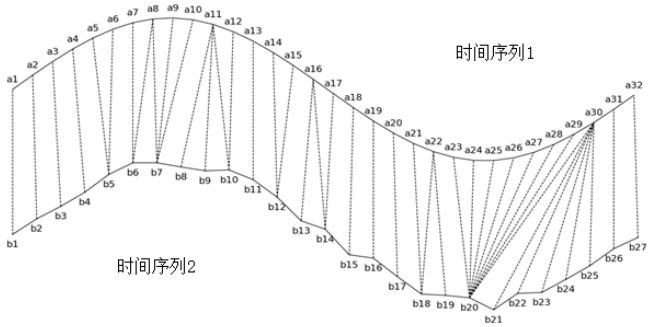


Figure 4. Time warping of two time series.

In the similarity calculation, the similarity between A and B is calculated as shown in Equation (1):

$$d_{ij} = \sum_{k=1}^{n} \sqrt{\left(a_i(x_k) - b_j(x_k)\right)^2 + \left(a_i(y_k) - b_j(y_k)\right)^2} \quad (1)$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

Where d_{ij} represents the total distance between the real-time trajectory and the standard trajectory. Based on Equation (1), the DTW distance between the two trajectory sequences can be derived, as shown in Equation (2):

$$DTW(X,Y) = min \sum_{i=1}^{n} \sum_{j=1}^{m} d_{ij} \quad (2)$$

Here, n is the length of time series A, and m is the length of time series B. Using Equation (2), the alignment distance between time series A and B can be calculated.

To evaluate how well a worker's current action matches the standard process, the system compares the real-time hand trajectory with each trajectory in the standard process trajectory library using DTW. Each standard trajectory in the library represents a correct assembly procedures pattern. By calculating the DTW distance between the real-time trajectory and each standard trajectory, the system generates a set of similarity scores. Let the real-time hand trajectory be sequence A, and the standard trajectory library contain k standard trajectories $\{B_1, B_2, \cdots, B_k\}$ After performing the DTW comparison between each standard trajectory and the real-time trajectory, a similarity score is generated:

$$Score_i = 100 - \rho \times DTW(A, B_i)^{\beta} \quad (3)$$

After normalization, the DTW distance is converted into a floating-point number between 0 and 1, where ρ and β are parameters that determine the mapping relationship between the DTW distance and the score. In this study, ρ is set to 92.4 and β is set to 0.97. $Score_i$ represents the similarity score between the i standard trajectory and the real-time trajectory; the higher the score, the more similar the two trajectories are. The final composite score is the average of all the similarity scores. This composite score is used to evaluate how well the worker's current operation matches the standard process. the higher the score, the more closely the worker's actions align with the standard process requirements.

2.2. Improved Dual-Channel Spatial Feature Fusion Network

In the entire approach, the key to ensuring detection accuracy lies in the precision and completeness of trajectory acquisition, which is directly dependent on the performance of the hand detection algorithm. Given the characteristics of fixed cameras and static backgrounds in assembly detection, this study designs a Hand Motion Sensing Module (HMSM) and a Differential-Attentional Feature Fusion Module (Diff-AFF) in the feature fusion network of YOLOv9, based on the traditional feature pyramid. These modules combine the RGB assembly images with the foreground images of hand movements, forming a dual-channel input for the YOLOv9 detection network. As shown in Figure 5, the process begins by using the Hand Motion Sensing Module to process the input images, filtering out background information and generating texture images with spatial features of hand movements (DiffVein images). These images are then combined with the feature maps from the backbone network to form the dual-channel input for the spatial sensing module. Next, the Diff-Attentional Feature Fusion Module merges shallow spatial features from the backbone network with motion spatial features from the texture images, thereby capturing global spatial information. This global spatial information is further combined with the deep semantic features, enhancing the model's accuracy in detecting hand movements in complex assembly environments.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

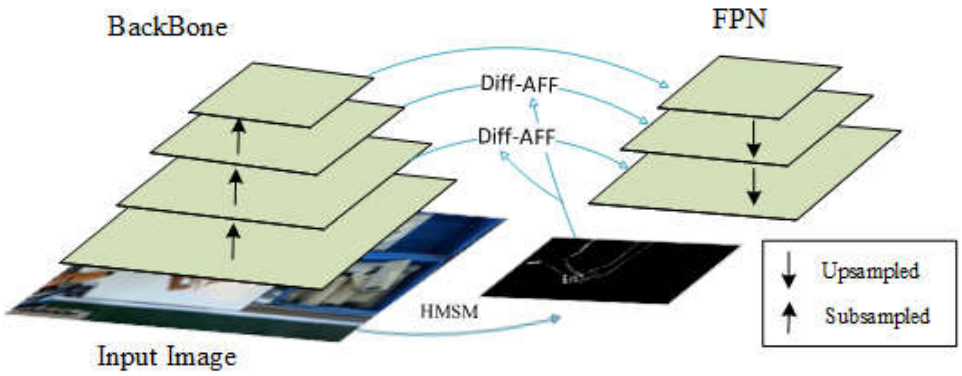


Figure 5. Improved dual-channel feature fusion network.

2.2.1. Palm Detection Based on Spatial Feature Fusion

The assembly process of workers in the workshop is shown in Figure 6. In the sequence of images, due to the interaction between the hand and tools, the pixels in the hand region undergo motion changes. Therefore, using pixel-based temporal differencing and the frame difference method, the texture information of the hand movement region can be extracted.

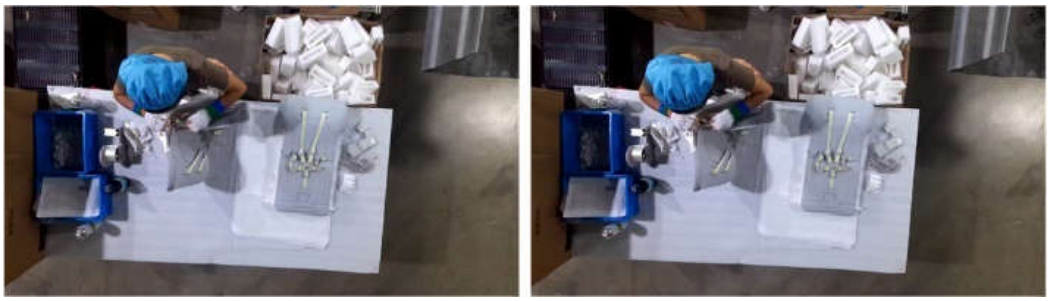


Figure 6. Frame images before and after the assembly process.

The frame difference method is calculated as follows: first, the pixel values of the corresponding RGB images in consecutive frames are subtracted to obtain a difference image. For two RGB images I_i and I_{i+1} taken from frame i and frame $i+1$, the RGB difference image I_{diff} s obtained using Equation (4):

$$I_{diff} = | I_{i+1} - I_i | \quad (4)$$

Next, the difference image is converted into a grayscale image I', removing the color information while preserving the texture, as shown in Figure 6(a). Then, a threshold segmentation method is used to separate the foreground and background in the grayscale image, filtering out background pixels. Threshold segmentation marks regions with pixel values greater than the threshold as foreground and those with values lower than the threshold as background, enhancing the contrast between the foreground and background. In this paper, the Otsu method, which is based on adaptive thresholding, is employed to segment the grayscale image I', and the result is shown in Figure 7.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

(a) Before the Otsu threshold (b) After the Otsu threshold

Figure 7. Effects before and after threshold segmentation.

The Otsu method searches for the threshold g_s that maximizes inter-class variance within the grayscale value range $g \in [0,255]$. The expressions for the threshold function δ and the inter-class variance σ^2 are as follows:

$$\delta(I'_{x,y}) = \begin{cases} 255, & I'_{x,y} > g_s \\ 0, & I'_{x,y} \le g_s \end{cases} \tag{5}$$

$$\sigma^2(g) = \psi_1(m_1 - m_G)^2 + \psi_2(m_2 - m_G)^2 \tag{6}$$

In these equations, m_1 and m_2 represent the mean grayscale values of the foreground and background pixels, respectively, and m_G is the global mean of the pixel values. ψ_1 and ψ_2 represent the probabilities of foreground and background pixels, respectively, when segmented by the threshold g. The probability distribution of a pixel with grayscale value i is $p_i = \frac{n_i}{N}$, where n_i is the number of pixels with grayscale value i in the image, and N is the total number of pixels in the image. When the threshold is set to g, the probabilities of the image being divided into foreground and background are $\sum_{i=0}^{g} P_i$ and $\sum_{i=g+1}^{255} P_i$. respectively, and the mean grayscale values of the foreground and background are given by:

$$m_1 = \frac{i = k+1}{\psi_1}, m_2 = \frac{i = k+1}{\psi_2} \tag{7}$$

Using the threshold function δ to filter out background information, the segmented texture image I_d is obtained and used as the dual-channel input for the object detection network, as shown in fig 6(b). The dense pixel areas in the image highlight the hand's movement trajectory. Region segmentation based on the frame difference method can effectively suppress background noise, providing spatial regions of interest for the detection network and guiding the accurate localization of occluded hands.

2.2.2. AFF-Attention Feature Fusion Module

To enhance the model's spatial perception of detection targets, this paper introduces the AFF (Attention Feature Fusion) module, which integrates the spatial information from texture images with the semantic features in the feature pyramid. The goal of feature fusion is to combine features from different layers or branches to fully exploit various image characteristics, achieving more robust and accurate target recognition. Although feature fusion is often implemented through summation or concatenation operations, these methods may not fully capture the interrelations between features.

The MS-CAM (Multi-Scale Channel Attention Module) extracts channel attention weights through two different scale branches: one branch uses Global Average Pooling to capture global feature attention, while the other uses point-wise convolution to extract local feature attention. During the fusion process, MS-CAM effectively combines the spatial features of hand movement with high-level semantic features, thus improving detection accuracy. The computational process of the MS-CAM module is as follows:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

$$\mathrm{L(X)} = B\left(\mathrm{PWConv}_2\left(\delta\left(\mathrm{B}\left(\mathrm{PWConv}_1(\mathbf{X})\right)\right)\right)\right) \tag{8}$$

Here, PWConv represents point-wise convolution, where $L(X)$ retains the same shape as the input features to preserve the detail information in low-level features. Given the global channel context g(X) and the local channel context L(X) , the refined features X' obtained through MS-CAM can be expressed as:

$$\mathrm{X}' = \mathrm{X} \otimes \mathrm{M(X)} = \mathrm{X} \otimes \sigma\left(\mathrm{L(X)} \oplus \mathrm{g(X)}\right) \tag{9}$$

In this equation, $\mathrm{M(X)}$ represents the attention weights generated by MS-CAM, σ denotes the activation function, $\oplus$ indicates element-wise addition, and $\otimes$ represents element-wise multiplication. Furthermore, when using MS-CAM to represent the AFF (Attention Feature Fusion), the formula is as follows:

$$\mathrm{Z} = \mathrm{M(X} \uplus \mathrm{Y)} \otimes \mathrm{X} + \left(1 - \mathrm{M(X} \uplus \mathrm{Y)}\right) \otimes \mathrm{Y} \tag{10}$$

In this equation, $Z \in R^{C \times H \times W}$ represents the fused features, and $\uplus$ denotes the initial feature integration. By introducing the MS-CAM module, the fusion of texture and semantic features is significantly improved, enhancing the ability to localize target areas and optimizing the performance of the object detection network. A schematic diagram of the AFF is shown in the Figure 8.

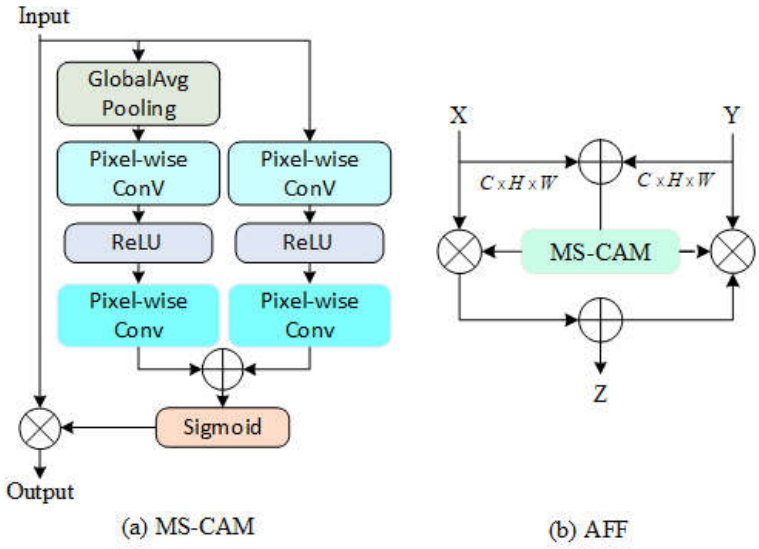


Figure 8. AFF - Attention Feature Fusion Module.

3. Results

To verify the accuracy of the proposed assembly detection solution in real production environments and its adaptability to different workstation assembly procedures, this study constructed an experimental dataset based on the assembly procedures collected from the partner companies, Ningbo Qixi Electric Co., Ltd. and Ningbo HOKO Electric Co., Ltd. The dataset covers video data from 10 assembly workstations, including three types of assembly outcomes: correct assembly, incorrect assembly, and missing assembly. A key factor affecting the proposed solution is the accuracy of hand detection in complex assembly environments. To validate the effectiveness of the background removal method based on the improved YOLOv9 object detection algorithm, 2000 pairs of images (including detection frames and background frames) were extracted from the videos and split into a training set and validation set in a 7:3 ratio for further analysis and evaluation of hand detection performance.

The experiments were conducted in a server-side environment with the configurations shown in Table 1. Both the original and improved models were trained for 150 epochs, with a batch size set

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

to 32. The Stochastic Gradient Descent (SGD) optimization algorithm was used, with an initial learning rate of 0.01, momentum set to 0.9, and a weight decay factor of 0.0005. During the training process, various data augmentation techniques were applied, such as random scaling and cropping, random rotation and flipping, random brightness adjustment, and mosaic image augmentation, to enhance the model's generalization capability.

Table 1. Experimental environment.

Class	Environmental	Class	Environmental
CPU	i7-12700 2.10 GHz	CUDA Version	CUDA 11. 2
GPU	GeForce RTX 4070	DL Framework	Pytorch
RAM	32 GB	Runtime	Linux
System	Ubantu 20.04.6	Script	Python3. 9

The evaluation metrics for model performance include Precision (P), Recall (R), Mean Average Precision (MAP), Floating Point Operations (FLOPs), and Frames Per Second (FPS)[29]. The calculation formulas for these metrics are as follows:

$$P = \frac{TP}{TP + FP} \tag{11}$$

$$R = \frac{TP}{TP + FN} \tag{12}$$

$$MAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{13}$$

In the formulas, TP represents true positives, FP represents false positives, and FN represents false negatives. The model accuracy is evaluated using Average Precision (AP), Mean Average Precision (MAP), Precision (P), and Recall (R). The real-time performance of the model is assessed through the number of frames detected per second (Frame/s or FPS).

3.1. Improved YOLO Results Analysis

To validate the performance of the improved algorithm, this study compares the inference speed and mAP metrics of traditional Faster R-CNN, YOLOv5s[30], YOLOv8s, YOLOv9s, and the proposed improved algorithm.

As shown in Table 2, the proposed YOLOv9s+Diff+AFF model demonstrated the highest accuracy in the assembly detection task, achieving an mAP@0.5 of 93.94%, which is an improvement of 10.02%, 9.62%, 5.99%, and 4.07% over Faster-RCNN, YOLOv5s, YOLOv8s, and YOLOv9s, respectively. Although the YOLOv9s+Diff+AFF model had a slightly lower frame rate (69.6 frames per second), its FLOPs were only 29.6B, and the model size was 16.43MB, indicating that it still maintains certain advantages in terms of computational cost and storage requirements. In comparison, YOLOv9s performed better in terms of lightweight design and recognition speed, but the improvements brought by the frame difference method and AFF module in the YOLOv9s+Diff+AFF model significantly enhanced detection accuracy in static background assembly scenarios. This makes it especially suitable for industrial assembly settings where high accuracy is required. Considering the characteristics of the workstation environment, the model achieves a good balance between real-time performance and detection accuracy.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

Table 2. Comparisons between this model and the mainstream models.

Model	mAP@0.5%	Frame/s	FLOPs(B)	Size/MB
Faster-RCNN	83.92	13.11	226.4	108.7
YOLOv5s	84.32	74.9	24.6	15.30
YOLOv8s	87.95	94.9	28.6	21.61
YOLOv9s	89.87	96.78	26.7	14.69
Ours	93.94	69.6	29.6	16.43

To validate the effectiveness of the two proposed improvement strategies and their performance in hand recognition in complex assembly environments, ablation experiments were conducted on the improved YOLOv9 dual-channel model, as shown in Table 3.

Table 3. Comparison of main module ablation experiments.

Model	Precision/%	Recall/%	mAP@0.5%	Frame/s
YOLOv9s	87.81	86.64	89.87	96.78
YOLOv9s+Diff	89.1	87.2	91.56	85.5
YOLOv9s+Diff+AFF	92.42	88.97	93.94	69.6

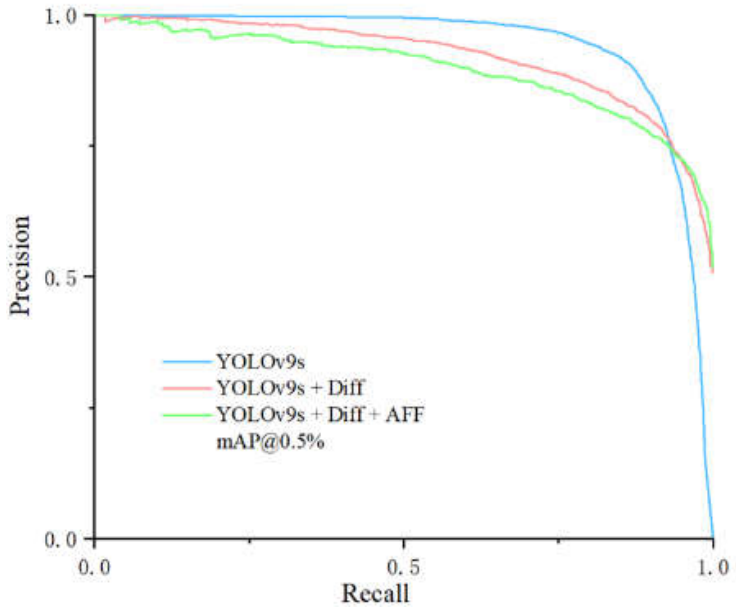


Figure 9. Comparison Curve of Precision Recall Improved Module

According to the latest experimental data presented in the table, ablation experiments demonstrated the hand recognition performance of the two proposed strategies in complex assembly environments. Compared to the baseline YOLOv9s model, YOLOv9s+Diff, which introduces the frame difference method to extract difference images in static backgrounds, showed improvements in both detection accuracy and recall rate. The mAP@0.5 increased from 89.87% to 91.56%, recall rate rose to 87.2%, and precision reached 89.1%, with the frame rate remaining at 85.5 frames per second, indicating that this strategy effectively leverages the static background characteristics to enhance hand position detection performance.

Building upon this, with the further introduction of the AFF module, the YOLOv9s+Diff+AFF model achieved an mAP@0.5 of 93.94%, precision improved to 92.42%, and recall rate increased to 88.97%. Although the frame rate dropped to 69.6 frames per second, the model's ability to detect multi-scale and dynamic targets was significantly enhanced. This indicates that the dual improvement strategy—combining difference images and the AFF module—further optimized the

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

model's detection accuracy and robustness in complex assembly environments, achieving a good balance between detection performance and real-time capabilities. The detection effects before and after the improvements are shown in Figure 10. By introducing motion information from consecutive images, the improved model captures pixel changes between frames, enhancing its ability to localize the hand in global space and reducing the impact of factors such as ambient lighting, hand-object occlusion, and similar background colors. It effectively resolves the issue of the network losing local key information when the hand touches the target, thereby significantly improving detection accuracy.

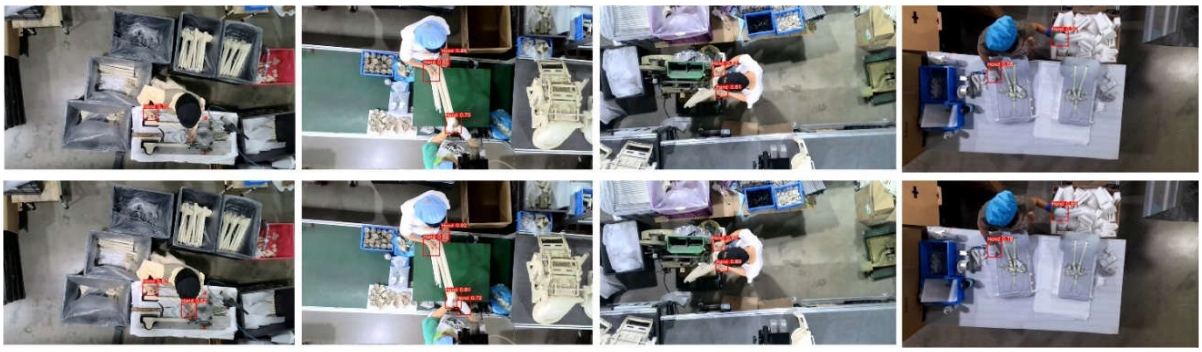


Figure 10. Detecting results before and after dual channel improvement.

3.2. Assembly Test Results Analysis

To validate the feasibility of the assembly detection method based on palm trajectories, we designed and conducted a series of experiments covering various complex assembly workstation conditions, including single-step and multi-step assemblies, as well as large and small part assemblies. The experiments involved eight different scenarios, each simulating typical assembly tasks from real production lines to test the system's compatibility and robustness in handling different worker operation habits and assembly methods. The system captures and analyzes palm trajectories in real time to detect the conformity of the assembly sequence with the standard process model and to identify missing or incorrect assemblies. To further assess the system's performance, we intentionally introduced missing and incorrect assembly events to simulate typical errors in actual operations, aiming to evaluate the system's detection accuracy and sensitivity to various errors across different assembly scenarios. As shown in Table 4 and Table 5, the experiments used fixed cameras for filming and the improved YOLOv9s network for real-time detection, thereby verifying the system's actual performance in assembly scenarios.

Table 4. Experimental Scenarios and Detection Performance Metrics.

	Number of Assembly personnel	Assembly Object Type	Participant Type	Missing Detection Accuracy (%)	Incorrect Detection Accuracy (%)
1	Single	Large	Skilled	98	97
2	Single	Small	Skilled	97	95
3	Multi	Large	Skilled	96	95
4	Multi	Small	Skilled	95	94
5	Single	Large	Novice	99	98
6	Single	Small	Novice	97	96
7	Multi	Large	Novice	96	95
8	Multi	Small	Novice	94	93

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

Table 5. Comparison of the effects of model improvements.

Model Version	Missing Accuracy (%)	Incorrect Accuracy (%)
YOLOv9s	90	91
ours	97	95

The experimental results show that the improved assembly detection system demonstrated excellent accuracy and real-time performance across different workstations. In detecting missing and incorrect assemblies, as well as assembly timing, the system achieved a high recognition rate in both small and large part assembly, as well as in single-step and multi-step scenarios. Specifically, the system's average detection accuracy for missing and incorrect assemblies across all workstations reached 96%, indicating high sensitivity to different types of errors. Moreover, the system exhibited strong cross-scenario adaptability, performing well in varying worker operation habits and environmental changes. This indicates that the system possesses good robustness and generalization capability, making it suitable for complex and variable assembly environments on production lines.Compared to the unmodified YOLOv9 model, the system optimized with the frame difference method and AFF module showed significant improvements in detecting hand positions and assembly procedures, with the average recognition rate increasing from 91% to 96%. Additionally, the system maintained a high response speed in fast-paced production lines, ensuring real-time feedback on assembly anomalies in practical applications. These experimental results validate the practicality and reliability of the palm trajectory-based assembly detection solution in industrial production.

4. Conclusions

This paper addresses the challenges of flexibility and complexity in manual assembly lines by proposing an adaptive assembly detection method based on hand movement trajectories and introducing innovative improvements to the YOLOv9 model, aimed at overcoming the limitations of traditional assembly detection methods. First, in terms of the assembly detection approach, this paper pre-defines assembly regions and transforms workers' hand trajectories into high-level abstractions of the assembly process, building a standard procedure model based on the assembly procedures of skilled workers. During the detection phase, the Dynamic Time Warping (DTW) algorithm is employed to perform real-time trajectory comparisons, effectively detecting missing and incorrect assembly actions.

Second, in terms of the hand detection algorithm, considering the fixed camera and static background characteristics, this paper extracts hand motion texture information using the frame difference method and introduces the Attention Feature Fusion (AFF) module to fuse multi-scale features with the difference images, thereby enhancing the target detection performance. The experimental results show that the improved YOLOv9s+Diff+AFF model achieves an mAP@0.5 of 93.94%, demonstrating significant improvements over traditional object detection algorithms such as Faster-RCNN, YOLOv5s, and YOLOv8s. Although the frame rate slightly decreased to 69.6 frames per second, it still meets the real-time detection requirements of assembly lines. The assembly detection system exhibited high detection accuracy and good generalization capability across various complex workstations and worker operations, achieving an average detection accuracy of 96%, validating the practical value of the proposed method.

Future research could further explore additional information contained in the trajectories to achieve more fine-grained assembly procedure detection, enhance the understanding of workers' assembly procedures, and enable more precise monitoring of the assembly process and error warning systems, further improving the intelligence level of manual assembly lines.

Author Contributions: Authors Shaoyi Zhou and Xiancheng Wang contributed to the development and conception of this research project. Authors Zewen Liu, Yifan Jiang and Lang Huang contributed to the preparation, collection, and provision of data, as well as the evaluation and interpretation of the data. The initial draft was written by Shaoyi Zhou and reviewed and revised by author Xiancheng Wang.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to commercial privacy or ethical restrictions.

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

References

1. Jimeno-Morenilla, A., Azariadis, P., Molina-Carmona, R., Kyratzi, S., & Moulianitis, V. (2021). Technology enablers for the implementation of Industry 4.0 to traditional manufacturing sectors: A review. Computers in Industry, 125, 103390.
2. Miqueo, A., Torralba, M., & Yagüe-Fabra, J. A. (2020). Lean manual assembly 4.0: A systematic review. Applied Sciences, 10(23), 8555.
3. Cohen, Y., Naseraldin, H., Chaudhuri, A., & Pilati, F. (2019). Assembly systems in Industry 4.0 era: a road map to understand Assembly 4.0. The International Journal of Advanced Manufacturing Technology, 105, 4037-4054.
4. Gu Xingchen.(2024). Research on Assembly Inspection of Engine Oil Pan System Based on Machine Vision (Doctoral dissertation, Jilin University)
5. Carvalho, P., Lafou, M., Durupt, A., Leblanc, A., & Grandvalet, Y. (2024). Detecting visual anomalies in an industrial environment: Unsupervised methods put to the test on the AutoVI dataset. Computers in Industry, 163, 104151.
6. Arjun, P., & Mirnalinee, T. T. (2016). Machine parts recognition and defect detection in automated assembly systems using computer vision techniques. Rev. Téc. Ing. Univ. Zulia, 39(1), 71-80.
7. Burresi, G., Lorusso, M., Graziani, L., Comacchio, A., Trotta, F., & Rizzo, A. (2021, June). Image-based defect detection in assembly line with machine learning. In 2021 10th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-5). IEEE.
8. Lü Nengbin, Wang Yaowei, Yang Min, Cao Zhenghong, & Du Fuzhou. A Virtual and Real Edge Matching Method for Missing Assembly Detection of Complex Product. Manufacturing Automation, 46(5), 36-42.
9. Huang, B., Liu, J., Zhang, Q., Liu, K., & Wang, J. (2023). Visual Detection Method for Missing Infusion Bag Pipeline. Electronics, 12(12), 2574.
10. Lu, Y., Xu, X., & Wang, L. (2020). Smart manufacturing process and system automation–a critical review of the standards and envisioned scenarios. Journal of Manufacturing Systems, 56, 312-325.
11. Wang Cheng, Huang Yichao, & Yang Guifeng. (2023). Workshop tool detection algorithm based on spatial feature fusion. Journal of Electronic Measurement and Instrumentation, 37(3), 39-49.
12. Zhao, S. W., Wang, J. F., Li, W., & Lu, L. F. (2023). Online assembly inspection integrating lightweight hybrid neural network with positioning box matching. IEEE Access.
13. Yang, Y., Wang, J., Liu, T., Lv, X., & Bao, J. (2020). Improved Long Short-Term Memory Network with Multi-Attention for Human Action Flow Evaluation in Workshop. Applied Sciences, 10(21), 7856.
14. Al-Amin, M., Qin, R., Moniruzzaman, M., Yin, Z., Tao, W., & Leu, M. C. (2023). An individualized system of skeletal data-based CNN classifiers for action recognition in manufacturing assembly. Journal of Intelligent Manufacturing, 1-17.
15. Koch, J., Buesch, L., Gomse, M., & Schueppstuhl, T. (2022). A methods-time-measurement based approach to enable action recognition for multi-variant assembly in human-robot collaboration. Procedia CIRP, 106, 233-238.
16. Li, W. (2021, March). Analysis of object detection performance based on Faster R-CNN. In Journal of Physics: Conference Series (Vol. 1827, No. 1, p. 012085). IOP Publishing.
17. Lu, X., Ji, J., Xing, Z., & Miao, Q. (2021). Attention and feature fusion SSD for remote sensing object detection. IEEE Transactions on Instrumentation and Measurement, 70, 1-9.
18. Chandana, R. K., & Ramachandra, A. C. (2022). Real time object detection system with YOLO and CNN models: A review. arXiv Prepr. arXiv2208, 773.
19. Wu Yibang, Chen Zhe, Li Zhe, Xiang Daxiang, & Cui Changlu. (2024). Remote sensing image detection method for illegal cultivation areas on steep slopes prohibited from reclamation based on improved YOLOv9. Transactions of the Chinese Society of Agricultural Engineering, 40(17).
20. Wang, C. Y., Yeh, I. H., & Liao, H. Y. M. (2024). Yolov9: Learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2402.13616.
21. Yang, S., Cao, Z., Liu, N., Sun, Y., & Wang, Z. (2024). Maritime Electro-Optical Image Object Matching Based on Improved YOLOv9. Electronics, 13(14), 2774.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2024 doi:10.20944/preprints202411.1140.v1

22. Sapkota, R., Meng, Z., Ahmed, D., Churuvija, M., Du, X., Ma, Z., & Karkee, M. (2024). Comprehensive Performance Evaluation of YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments. arXiv preprint arXiv:2407.12040.
23. An, R., Zhang, X., Sun, M., & Wang, G. (2024). GC-YOLOv9: Innovative smart city traffic monitoring solution. Alexandria Engineering Journal, 106, 277-287.
24. Chien, C. T., Ju, R. Y., Chou, K. Y., & Chiang, J. S. (2024). YOLOv9 for fracture detection in pediatric wrist trauma X-ray images. Electronics Letters, 60(11), e13248.
25. Chen, Y., Du, Z., Li, A., Li, H., & Zhang, W. (2024, June). Substation Equipment Defect Detection Based on Improved YOLOv9. In 2024 IEEE 4th International Conference on Software Engineering and Artificial Intelligence (SEAI) (pp. 87-91). IEEE.
26. Zou, J., & Wang, H. (2024). Steel Surface Defect Detection Method Based on Improved YOLOv9 Network. IEEE Access.
27. Dai, Y., Gieseke, F., Oehmcke, S., Wu, Y., & Barnard, K. (2021). Attentional feature fusion. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 3560-3569).
28. Wei, Qiuyue, & Liu, Yufan. (2021). Dynamic gesture recognition based on Kinect and improved DTW algorithm. Sensors and Microsystems.
29. Yacouby, R., & Axman, D. (2020, November). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the first workshop on evaluation and comparison of NLP systems (pp. 79-91).
30. Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Michael, K., ... & Mammana, L. (2022). ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. Zenodo.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.