Online assembly inspection integrating lightweight hybrid neural network with positioning box matching

Shiwen Zhao
Huazhong University of Science and Technology

Junfeng Wang (wangjf@mail.hust.edu.cn)
Huazhong University of Science and Technology

Wang Li
Huazhong University of Science and Technology

Longfei Lu
Huazhong University of Science and Technology

Research Article

Keywords: Assembly inspection, Lightweight hybrid neural network, Vision transformer, Positioning box matching

Posted Date: May 24th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2964378/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.
Online assembly inspection integrating lightweight hybrid neural network
with positioning box matching

Shiwen Zhao, Junfeng Wang*, Wang Li, Longfei Lu

Department of Industrial and Manufacturing System Engineering, School of Mechanical Science and Technology,
Huazhong University of Science and Technology, Wuhan 430074, China

*Corresponding author. E-mails: wangjf@mail.hust.edu.cn

Abstract: Assembly inspection methods have been widely used in the process of mechanical product assembly for quality issues. However, some challenges remain to be solved, such as low detection efficiency, poor accuracy and sensitive to camera view. This paper proposes an online assembly inspection scheme based on lightweight hybrid neural network and positioning box matching. A lightweight hybrid neural network is proposed to simultaneously detect key points and parts with high accuracy and strong robustness. Utilizing the key points detection results, the transformation relationships between the in-site assembly images and the standard templates are solved. According to the results of assembly parts detection, the detected 2D positioning bounding boxes are matched with those in the standard assembly templates, so as to evaluate whether the current step has quality problems. In addition, the proposed method is tested on an assembly dataset constructed in this paper. For key points detection, the average error is less than 1 pixel. For parts detection, the mean average precision is 97.66%. The missing and wrong assembly inspection results show that the average F1 score reaches 93.96%. This inspection method can be employed to detect the missing and wrong assembly errors of each assembly step online, improving the assembly quality of products.

Key words: Assembly inspection, Lightweight hybrid neural network, Vision transformer, Positioning box matching

1 Introduction

Assembly process is a key stage of mechanical products development. In modern industrial production, the demand for small batch customized products is increasing rapidly. Due to economic and technical reasons, small batch complex products tend to be produced in the environment of manual assembly. However, in this type of production, product quality and assembly efficiency largely depend on the skills of workers [1]. High product complexity and diversity significantly increase the cognitive load of employees, increasing the probability of assembly faults [2]. Hence, an assembly inspection process needs to be set to inspect if operators complete the assigned task correctly or not.
The commonly used method of assembly faults inspection is to manually compare the assembled products with 2D process manual or 3D model, checking whether there are missing components and whether the type, position or angle of assembly parts reach the process requirements. As most of the mechanical parts are single in color, lack of texture and have various types, assembly inspection is a time-consuming and laborious work. At the same time, the classical method lacks detection and control for each assembly step. When quality problems such as missing or wrong assembly occur, parts need to be disassembled again, which seriously affects assembly efficiency. Therefore, online quality inspection system has been gradually applied in the assembly field to detect and avoid errors in real time, thus ensuring consistent product quality and improving efficiency [3].

Currently, vision-based inspection methods are used by more and more researchers. It is simple and efficient, without additional introduction of some complex sensors, and can adapt to most mechanical products [4]. The traditional visual detection method based on template matching adopts pixel-wise processing to extract and describe features of the real assembly images and match them with standard assembly templates. It can accomplish the task well under the condition of rich color and texture of assembly scene and simple background. However, due to the large search space of template matching, the detection speed of this method is slow, and it does not perform well in dynamic assembly scenes with complex background, changeable camera views and lighting condition. Moreover, in most cases, mechanical parts are consistent in color, lack of texture, making feature extraction and description difficult. Recently, as deep learning has made remarkable achievements in the field of image processing, more and more researchers begin to explore its application in assembly inspection [5]. Powerful deep neural network can extract and analyze image features from complex assembly in-site images and detect errors with relatively high speed and accuracy. Nevertheless, the current vision inspection methods focus more on the missing assembly of parts, and there are few researches on the inspection of wrong assembly problems, such as assembly position or angle error. At the same time, in the assembly process, the relative position of the camera and parts will also have an important impact on the inspection results.

Aiming to further address these issues, this paper proposes an innovative assembly inspection method based on deep learning and positioning box matching to inspect the missing and wrong assembly problems in each assembly step of the product. The main contributions of this paper lie in two aspects: (1) An encoder-decoder model integrating the Transformer attention mechanism, termed CNN-Transformer, is designed to integrate key points and parts detection with high accuracy, robustness and timelines. (2) A homography matrix estimation method based on key points detection is proposed to calculate the perspective difference between the in-site images and the standard assembly templates. Meanwhile, the detected positioning boxes containing rotation information
are matched with the template to determine whether the current assembly step has quality problems.

2 Literature review

2.1 Template matching methods for image based assembly inspection

In the literature, template matching methods are commonly used in assembly inspection. Liu et al. [6] studied a template based inspection technology in semi-closed narrow space. The Canny edge matching algorithm [7] and subpixel algorithm [8] were used to detect whether there were missing small parts and whether the assembly clearance met the requirements. Kim et al. [9] studied the detection of ship assembly quality. The GrabCut algorithm [10] was used to segment the ship section objects. Cojocaru et al. [11] studied an assembly quality inspection method based on image segmentation, which segmented assembly site images to obtain contour information of part areas.

The above methods can complete the inspection task well in certain cases. But when the camera view changes, there is a great difference between the actual images and the templates, so the inspection result is not ideal. To solve this, SIFT [12], SURF [13] and ORB [14] algorithms are commonly used to extract feature points in images, aligning the in-site image with template image. However, these methods require rich textures in target areas, which is not ideal in some texture-less assembly scenes. Hence, some researchers studied the edge feature based inspection methods. Yang et al. [15] proposed an edge based hatch cover category detection method. A fast cover edge feature and descriptor are designed to recognize different types of cover. Tsai et al. [16] utilized the Canny edge detection and the designed expectation-maximization algorithm to inspect the exact position of PCB components.

2.2 Deep learning methods for image based assembly inspection

2.2.1 Deep learning methods for object detection

Deep learning has significant advantages in image processing and has certain invariance in image geometric transformation, deformation and illumination. The object detection network based on deep learning mainly includes the Convolutional Neural Network (CNN) and Transformer.

Object detectors based on CNN have shown unique advantages on large scale image datasets [17]. Ren et al. [18] proposed a two-stage algorithm called Faster R-CNN. The recommended candidate regions were first generated, and the target detection and localization were then performed. To avoid the time-consuming process brought by two stage detection, Redmon et al. [19] proposed a single stage detector, YOLO, to meet the real-time
requirements. In the process of the above network detection, a large number of candidate boxes would be generated at the same target position. Therefore, Non-Maximum Suppression processing was required [20]. To avoid complex post processing, Zhou et al. [21] proposed Centernet, an object detection model based on Fully Convolutional Network (FCN) [22], where each target was taken as its single central point, so the Non-Maximum Suppression was not required.

In recent years, researchers have introduced Transformer architecture from the field of natural language processing into computer vision tasks. Transformer is a deep neural network structure based entirely on attention mechanism. Carion et al. [23] first introduced the Transformer architecture to object detection. The neural network consisted of three parts: a CNN backbone network, an encoder-decoder Transformer, and a detection head. Dosovitskiy et al. [24] applied a pure transformer structure, of which images were segmented into small pieces, and a linear embedded sequence of these small pieces was fed to a designed Transformer. Later, some researchers improved the transformer based visual detection method in terms of reducing the training difficulty and the number of parameters [25].

CNN and Transformer have their own advantages in processing visual information. CNN has strong rotation and translation invariant by using convolution and pooling operations, which is conducive to the effective identification and classification of parts. Meanwhile, features are extracted by sharing convolution kernel, avoiding redundant calculations. Transformer has a larger feature receptive field and stronger high-level semantic feature extraction capability, enabling more accurate target positioning. However, the fully Transformer-based model has a high computational complexity, which is caused by a significant amount of attention matrix calculations.

2.2.2 Deep learning methods for assembly inspection

At present, many researchers have studied assembly detection methods based on CNN. Andrés et al. [26] studied the movement detection of the operator, and SSD algorithm was used to conduct real time monitoring. To solve the assembly quality detection problem of explosion-proof lamp tubes in manual assembly, Riedel et al. [27] designed a system based on YOLOV4, which inspected the assembly omission errors and recorded them, ensuring the traceability of each step. Chen et al. [28] proposed an assembly monitoring method, where the depth images are employed to effectively detect the newly assembled parts from different perspectives.

The above mentioned researches mainly focus on the inspection of missing assembly errors, ignoring the wrong assembly errors caused by assembly type and assembly position or angle. Moreover, the relative pose changes of camera and object during the assembly will also make the inspection process difficult.
3 Lightweight hybrid neural network and positioning box matching based assembly inspection approach

3.1 Framework of the proposed inspection method

In this paper, we propose a deep learning and positioning box matching-based method to realize the online inspection of missing and wrong assembly problems. An encoder-decoder hybrid neural network, CNN-Transformer, is designed to locate and identify parts and to simultaneously detect key point coordinates on the assembly object. The assembly detection framework proposed in this paper is shown in Figure 1, which mainly includes offline preparation stage and online detection stage.

In the offline stage, training datasets is established according to the inspection task. It includes collecting assembly site pictures, labeling key points on assembly object and labeling positioning boxes of parts. The CNN-Transformer model is trained via images and labels in the datasets. At the same time, in the offline stage, the standard assembly templates of each step are collected by virtual camera in CAD. The 2D positioning boxes of the parts and key points are also marked for the subsequent object matching.

The task of the online inspection stage is to assess the assembly quality of each assembly step. positioning boxes for parts and coordinates of key points in in-site image are detected by the trained CNN-Transformer model. The detected key points are used to solve the transformation relation between standard template and in-site image. According to the transformation relation, the positioning box matching algorithm is employed to match boxes pre-marked in the template with those detected by the CNN-Transformer model. According to the matching results, it can be judged whether there are missing or wrong assembly problems in the current assembly step.
3.2 Integrated key point and part prediction model

3.2.1 Architecture of the proposed CNN-Transformer neural network

Centernet [21] is an anchor-free object detection network. In the object detection task, it models the object as a single point and generates heatmap to predict the location and size of the object. Inspired by Centernet, the CNN-Transformer is proposed in this paper to integrate the task of key point prediction and part positioning box prediction. As shown in Figure 2, the proposed CNN-Transformer model includes two parts: encoder and decoder. The input is a RGB image with a size of $512 \times 512 \times 3$, and the output is key point information and positioning box information.

The task of the encoder is to use the backbone integrated with CNN and Transformer to downsample the input image and extract semantic features. The output of the encoder is a $16 \times 16 \times 96$ feature map. Then the decoder part is used to reconstruct the feature map. In the decoder part, the channel dimension of the feature graph is firstly adjusted by two 1x1 convolution. To realize the prediction from pixel to pixel, deconvolution operation is then employed to up-sample the feature map for three times. A feature map with size $128 \times 128 \times 64$ is finally obtained. Finally, according to our detection task, the key point detection head and part detection head are set as output layers respectively. They share the weight parameters except for the detection head, forming a multi-task learning network, which greatly saves network parameters and processing time.

For the key point prediction head, two parallel convolution layers are used as the output layer to predict the center and offset heatmap of the key points respectively. The output size of the center heatmap is $128 \times 128 \times 4$, where four different channels are used to predict four different key points. To reduce the coordinate error caused by feature downsampling operation, we set another branch to regress the coordinate offset of each key point as compensation. The output size of the offset heatmap is $128 \times 128 \times 2$, and 2 is the two offsets of the key point in the width and height direction.

The ground truth of each key point is represented by a Gaussian circle, with the coordinate of the key point as the center of the circle. The Gaussian distribution is used to generate the values within the circle, and the values outside the circle are set to 0. The Gaussian probability distribution of the key point is calculated as follows:

$$P_{xyc} = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{(x-u_x)^2+(y-u_y)^2}{2\sigma^2}\right)$$

(1)

where $(u_x, u_y)$ is the coordinate of the key point, $(x, y)$ is the coordinate in the Gaussian circle. $\sigma$ is the variance, where 1/3 of the radius is taken.
For the prediction head of the positioning box, three parallel convolution layers, Center, WH and Angle, are used as the output layer to predict the heatmap of the center, size and rotation angle of the positioning box. The size of the Center heatmap is $128 \times 128 \times D$, where $D$ is the number of types of the parts. Similar to the key point prediction task, the ground truth of the Center heatmap is also represented by a Gaussian circle. The WH heatmap branch represents the width and height of the detection box, and its size is $128 \times 128 \times 2$.

The commonly used horizontal positioning box contains the position and the approximate size information of the part. During the assembly process, when the parts are tilted or rotated, it is hardly to fit the part well, especially for those parts with large aspect ratio. Considering that, we add the rotation information to make the detected box more fit the part. The rotation information is defined as the angle between the longer side of the part and the $y$-axis in this paper. Figure 3 shows the angle definition of the rotated positioning box. The range of angle is defined as $[0, \pi)$. 

Figure 2 Architecture of the CNN-Transformer neural network

Figure 3 Rotated positioning box
Finally, to train the CNN-Transformer detector, for the offset, weight, height and angle branch prediction, the $L_1$ loss is used. To solve the problem of unbalanced part category, the penalty-reduced regression with focal loss is employed to calculate the loss of key points prediction and the central point of the bounding box. The expression is as follows:

$$L_s = \frac{-1}{M} \sum_{xyc} \left\{ \begin{array}{ll}
(1 - Y_{xyc})^\alpha \log(Y_{xyc}) & \text{if } Y_{xyc} = 1 \\
(1 - Y_{xyc})^\beta \log(Y_{xyc}) + \alpha \log(1 - Y_{xyc}) & \text{otherwise}
\end{array} \right.$$  \hspace{1cm} (2)

where $M$ is the number of key points in one training image, and $\alpha$, $\beta$ are hyper parameters, which are set to 2 and 4 respectively. $Y_{xyc}$ and $Y_{xyc}^\hat{\cdot}$ are the actual and predicted value of key point.

### 3.2.2 Hybrid encoder combining CNN and Transformer

Mechanical parts are similar in color, lack of texture and may have changes of rotation and translation during the assembly process, which brings challenges to the feature extraction, location and recognition of parts. To improve the accuracy of parts detection, the transformer attention mechanism is integrated into the MobileNetV2 [29] backbone as an improved feature encoder.

As shown in Figure 2, the encoder is mainly made up of a stack of Conv block and Vit block. The Conv block extract features through CNN layer, and the same residual block structure as MobileNetV2 is adopted. The Vit block is mainly made up of transformer, which is placed in the back half of the encoder to extract higher level semantic features. Due to the high computational complexity of transformer in processing large-size feature map, we alternate the CNN layer with the Transformer block. Therefore, the CNN layer is mainly used to down sample the feature map, reducing the computational cost of the Transformer.

The Vit block is shown in Figure 4. For a $H \times W \times C$ feature map, patch embedding is applied first. A pointwise convolution is used to project the dimension of the feature map to a higher dimension $H \times W \times d$. The feature map is then expanded with the dimension of $(h \times w) \times N \times d$. Then, it is fed into the Transformer structure for feature extraction.
Transformer structure consists of multi-head self-attention (MSA) and feed forward modules, with residual connection and layer normalization applied after each block. The feed forward module is a multilayer perceptron block that projects the dimension of the feature map to four times the original dimension and then projects back to the original dimension. For each feature map whose input dimension is $N \times d$, as shown in Figure 5, the self-attention module obtains three quantities $Q$ (Query), $K$ (key) and $V$ (value) by multiplying them with three learnable matrices, and then calculates the final self-attention output using equation (3).

$$Self - Attention(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

where $d_k$ is the scale factor and its value is the variance of the $Q \cdot K^T$.

To allow transformer to notice information in different subspaces and capture richer features, Multi-head self-attention is used. Multiple groups of learnable matrices $W^Q$, $W^K$ and $W^V$ are set to multiply with the input feature map. The results are concatenated together, and the output dimension is projected to the input dimension by equation (4).

$$MSA(Q,K,V) = \text{Concatenate}(head_1,head_2,...,head_h)W^o \quad (4)$$

where $head_i$ is the output of each single-head attention and $W^o$ is the linear projection operation.

After inputting $w \times h$ feature maps with $N \times d$ dimensions into transformer for processing, the dimension of the obtained feature maps is still $(h \times w) \times N \times d$. Then, the feature maps are folded into $H \times W \times d$ and projected back to the original $H \times W \times C$ dimensions through a pointwise convolution.

**Figure 5** Self – Attention module
3.3 Transformation matrix calculation and positioning box matching

During assembly process, the position or angle of the assembly object on a workbench usually changes with workers’ operations. Hence, there is a difference in perspective between the pre-collected standard assembly template and the in-site image, which will cause error for the subsequent positioning box matching to inspect assembly problems. To align the in-site image and the standard template, at least four points corresponding to the in-site image and the template located in the same plane are needed to solve the transformation matrix according to the perspective transformation theory. The type and location of the four key points in the standard assembly template can be labeled and measured in advance. Meanwhile, the coordinates of the four key points in the in-site image plane can be predicted by the proposed CNN-Transformer model. The maximum value in the key point heatmap output by CNN-Transformer is selected as the pixel coordinate of a key point, and four channels of the center heatmap are responsible for detecting four different key points respectively. Based on the detected four key points from the in-site image and the corresponding key points pre-set in the standard template, the transformation matrix can be calculated by the following formula:

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  h_{11} & h_{12} & h_{13} \\
  h_{21} & h_{22} & h_{23} \\
  h_{31} & h_{32} & h_{33}
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

where \((x, y)\) is the key point coordinate predicted by the CNN-Transformer network from in-site image, and \((x', y')\) is the corresponding coordinate on the standard assembly template.

Based on transformation matrix, the in-site image is aligned with the template. Then the error of missing and wrong assembly is inspected by the positioning box matching algorithm. The matching algorithm calculates the intersection over union (IoU) between the detected box and the template box to determine whether the corresponding relation between the actual assembly position and the standard position is correct. Figure 6 shows the actual positioning box and standard template box of the current assembly part. The IoU matching formula is as follows:

\[
IoU_{2D} = \frac{S_r \cap S_v}{S_r \cup S_v}
\]

where \(S_r\) is the bounding box in the standard assembly template, and \(S_v\) is the actual detection box after the homography transformation.
The positioning box matching algorithm contains the following steps.

Step 1: Select a part assembled in this step from the template, named $A$.

Step 2: The standard positioning box of $A$ is expressed as $(O, P_0, P_1, P_2, P_3)$, where $O$ is the central point and $P_0, P_1, P_2, P_3$ is the four vertices. Search the positioning box of the part intersecting with $A$ at the corresponding position of the in-site image, denoted as $A_i$. If it is not searched, the current step exists a missing assembly problem.

Step 3: According to equation (6), calculate the IoU of $A$ and $A_i$. If the IoU is less than the threshold value, it is considered that the current step exists a wrong assembly problem.

Step 4: Judge the category of $A$ and $A_i$. If they are of the same type, this step can be considered as qualified assembly. Otherwise, the current step exists an assembly type error problem.

Step 5: Repeat the above steps for all parts assembled under the current assembly step in the template.

In this paper, the positioning box matching threshold is set to 0.65. When the IoU matching result is less than the threshold, it is considered that the current step has an unqualified assembly problem.

4 Experiments and analysis

4.1 Template obtaining and neural network training

The public assembly model, MONA [30], is used for inspection experiments. MONA is a bottom plate based assembly with a length, width and height of $250 \times 150 \times 50$ mm, consisting of 16 parts and 18 assembly steps. We use 3D printing to process corresponding assembly object parts. Eight main parts in MONA are selected for the subsequent experiments, namely TR_M6_Corner, BR_Column, Square, Cylinder, TM_Attachment, TL_M4_Hole, TL_M4_Rest and BL_Block_Top, which consists of 10 assembly steps in total. For template images obtaining, we use the virtual camera in CAD to collect images of each assembly step in the standard assembly state. For each template, the key points and standard positioning box information are manually marked. Partial template images
and annotated information are shown in Figure 7.

![Figure 7](image1.png)  
**Figure 7** Template obtaining for partial assembly steps

To train the CNN-Transformer network, 240 assembly images in the ten selected assembly steps are collected as the training set of MONA assembly, and 60 images are collected as the test datasets. These images are taken from different angles and under different lighting conditions in real assembly backgrounds, as shown in Figure 8. Each image is manually labeled with key points and positioning boxes information. To improve the generalization ability of the model, we use four common data enhancement methods, including image rotation, image flip, adding noise, and color space transformation. The final number of images in training datasets is 1200.

![Figure 8](image2.png)  
**Figure 8** Example of image data for training

The neural network training and subsequent test experiments are performed on a Windows 10 operating system, with an Intel Xeon e5-2630 CPU, 64 GB of RAM, and an NVIDIA Quadro P4000 graphics card. The initial learning rate is set to 1e-3. The batch size, weight decay, momentum is set to 4, 5e-4, 5e-4 respectively. The learning rate drops to one tenth of the original rate every 50 rounds, with a minimum learning rate of 1e-5.

4.2 Key points detection and error analysis

4.2.1 Performance of key points detection

To verify the effectiveness of the CNN-Transformer based assembly inspection method for key points
detection, four points, $P_0, P_1, P_2$ and $P_3$ on the base plate of MONA assembly are selected, as shown in Figure 9(a). During the assembly operation, the selected four points should not be occluded, thus ensuring that the algorithm can correctly solve the transformation relation between the assembly template and in-site image. The detected key points heatmap and result are shown in Figure 9(b) and Figure 9(c). As shown in Figure 9(b), the channel dimension of the output heatmap is 4, corresponding to 4 key points.

![Key points selected on the base plate](image1)

![Key points heatmap](image2)

![Key points prediction result](image3)

**Figure 9** Key point detection.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Key points detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key point</td>
<td>$x$</td>
</tr>
<tr>
<td>$P_0$</td>
<td>172.5</td>
</tr>
<tr>
<td>$P_1$</td>
<td>224.2</td>
</tr>
<tr>
<td>$P_2$</td>
<td>344.6</td>
</tr>
<tr>
<td>$P_3$</td>
<td>365.2</td>
</tr>
</tbody>
</table>

**4.2.2 Error evaluation of key point detection**

During the process of assembly, the rotation angle will change with the moment of the assembly object on the workbench. Meanwhile, due to the change of the distance between the camera and the assembly object, contents on the in-site image also have different scales. Hence, different rotation angles and grade of scale variations of the assembly object will affect the accuracy of key points detection. In this section, we compare the proposed CNN-Transformer method with SIFT, SURF, ORB and SuperPoint [31] methods, which are commonly used in key points detection. The SIFT, SURF and ORB are traditional feature points detection methods based on descriptors, while SuperPoint is based on deep learning. The Root mean square error (RMSE) is adopted to calculate the pixel error.
\[ RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (P_i - \bar{P}_i)} \]  \hspace{1cm} (8)

where \( N \) is the number of key points, \( P_i \) is the coordinate of the detected key point and \( \bar{P}_i \) is the ground truth.

For the influence of different rotation angles, as shown in Figure 10, eight images are collected, and the center of MONA rotates 0°, 20°, 40°, 60°, 80°, 100°, 120°, and 160° in the counterclockwise direction respectively. To reduce the influence of other factors, the image content collected only contains the baseplate of MONA, and no other interferences are set.

![Figure 10 Different rotation angles.](image)

The average pixel error of the four points obtained by each detection algorithm is shown in Figure 11. For different rotation angles, the key points detection errors of our method are relatively stable. The mean pixel error is about 1.5 pixels. SuperPoint shows high accuracy when the rotation angle is less than 60 degrees, and its error is about 0.5 pixels. But when the rotation angle is greater than 60 degrees, its detection error increases rapidly, about 6 pixels. Similarly, the ORB has poor accuracy and stability for different rotation angles, with a maximum error of about 7 pixels when the rotation angle is greater than 80 degrees. SIFT and SURF have relatively stable accuracy, and their results are about 1 pixel at different rotation angles. In conclusion, our algorithm has high detection accuracy under different rotation angles, which is superior to ORB based on descriptors and SuperPoint based on FCN.
To compare the scale invariance of different algorithms, Gaussian blur is applied to the acquired images to simulate the scale variation of the images caused by the camera shooting distance. As shown in Figure 12, the grade of scale variation is controlled by the size of Gaussian convolution kernel. The larger the convolution kernel is, the more blurred the image is, which means that the camera is farther away from the assembly object. The convolution kernel varies from 1 to 11, with a step size of 2, and is divided into 6 levels in total.

The average pixel error at different scales is shown in Figure 13. When the roughness of the image increases, the proposed algorithm and SurperPoint show good stability. For different grades of roughness, the pixel error of the proposed method is stable at 1.5 pixels, and the SuperPoint is stable at about 0.9 pixels. For the descriptor-
based detection method, the error increases with the increase of image roughness level. The image pixel error of SIFT and ORB is up to 150 pixels, and SURF is also over 120 pixels.

![Graph showing pixel error at different grade of scale variations](image-url)

**Figure 13** Pixel error at different grade of scale variations

The detection speed has an important influence on whether it can provide real time feedback of assembly inspection for operators. The average time and frame rate of these algorithms are also tested, and the result is shown in **Table 2**. The methods based on descriptors are more time-consuming, while the deep learning methods are fast. The frame rate of the algorithm proposed in this paper can reach 35.7, which can detect key points in real time.

<table>
<thead>
<tr>
<th>Algorithm type</th>
<th>ORB</th>
<th>SIFT</th>
<th>SURF</th>
<th>SuperPoint</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time consumption (ms)</td>
<td>112.74</td>
<td>271.54</td>
<td>185.10</td>
<td>34.50</td>
<td><strong>28</strong></td>
</tr>
<tr>
<td>Frame rate (FPS)</td>
<td>8.87</td>
<td>3.68</td>
<td>5.40</td>
<td>28.98</td>
<td><strong>35.7</strong></td>
</tr>
</tbody>
</table>

**4.3 Assembly parts detection**

**4.3.1 Comparison with other detection methods**

To verify the part detection performance of the CNN-Transformer model proposed in this paper, the 60 images of MONA assembly in the test datasets are selected. These parts have similar colors, different sizes and lack of texture. Partial detection results are shown in **Figure 14**. The 60 images in the test datasets are used to quantify the detection results, and the CNN-Transformer is compared with two types of mainstream algorithms in parts detection, including the Faster-RCNN, YoloV3 and SSD based on anchor and Centernet without anchor.
The results of various algorithms are shown in Table 3, which includes the detection accuracy, the number of model parameters, the detection time and frame rate. The detection accuracy is evaluated by the commonly used mean average precision (MAP). In terms of detection accuracy, the proposed CNN-Transformer achieves the highest MAP, and the algorithm has the lowest number of parameters. This means that our model is more lightweight and requires less computing resources, which is ideal for deployment on edge devices with limited computing resources. In terms of detection speed, SSD achieves the fastest detection speed, but our algorithm is only 2.42ms slower. The computational complexity of the model is also measured by the index of floating point operations (FLOPs). As shown in Table 3, the computational complexity of our model is the lowest in the compared ones.

### Table 3 Performance of parts detection

<table>
<thead>
<tr>
<th>Method</th>
<th>MAP0.5</th>
<th>Params (M)</th>
<th>Time (ms)</th>
<th>FPS</th>
<th>FLOPs (G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster-RCNN</td>
<td>95.56%</td>
<td>521.97</td>
<td>169</td>
<td>5.92</td>
<td>184.94</td>
</tr>
<tr>
<td>YOLOV3</td>
<td>83.63%</td>
<td>40.23</td>
<td>35.15</td>
<td>28.45</td>
<td>32.82</td>
</tr>
<tr>
<td>SSD</td>
<td>90.51%</td>
<td>93.64</td>
<td><strong>22.90</strong></td>
<td><strong>43.6</strong></td>
<td>24.54</td>
</tr>
<tr>
<td>Centernet</td>
<td>94.73%</td>
<td>124.61</td>
<td>24.50</td>
<td>40.82</td>
<td>35.10</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>97.66%</strong></td>
<td><strong>19.54</strong></td>
<td><strong>25.32</strong></td>
<td><strong>39.49</strong></td>
<td><strong>13.12</strong></td>
</tr>
</tbody>
</table>

#### 4.3.2 Experiments for the improved hybrid encoder

To verify the effectiveness of the improved hybrid encoder, the proposed hybrid encoder is compared with the pure Transformer or CNN encoders, including the Swin-Transformer [25] and Resnet50 [29], and the original MobileNetV2 [29] is also leveraged as the baseline. The experimental results are shown in Table 4.

### Table 4 Performance of parts detection with different backbone

<table>
<thead>
<tr>
<th>Method</th>
<th>MAP0.5</th>
<th>Params (M)</th>
<th>Time (ms)</th>
<th>FPS</th>
<th>FLOPs (G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster-RCNN</td>
<td>95.56%</td>
<td>521.97</td>
<td>169</td>
<td>5.92</td>
<td>184.94</td>
</tr>
<tr>
<td>YOLOV3</td>
<td>83.63%</td>
<td>40.23</td>
<td>35.15</td>
<td>28.45</td>
<td>32.82</td>
</tr>
<tr>
<td>SSD</td>
<td>90.51%</td>
<td>93.64</td>
<td><strong>22.90</strong></td>
<td><strong>43.6</strong></td>
<td>24.54</td>
</tr>
<tr>
<td>Centernet</td>
<td>94.73%</td>
<td>124.61</td>
<td>24.50</td>
<td>40.82</td>
<td>35.10</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>97.66%</strong></td>
<td><strong>19.54</strong></td>
<td><strong>25.32</strong></td>
<td><strong>39.49</strong></td>
<td><strong>13.12</strong></td>
</tr>
<tr>
<td>Backbone</td>
<td>MAP(_0.5)</td>
<td>Params (M)</td>
<td>Time (ms)</td>
<td>FLOPs (G)</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>------------</td>
<td>----------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>Swin-Transformer</td>
<td>98.39%</td>
<td>27.49</td>
<td>43.46</td>
<td>22.85</td>
<td></td>
</tr>
<tr>
<td>Resnet50</td>
<td>94.12%</td>
<td>23.51</td>
<td>25.30</td>
<td>20.37</td>
<td></td>
</tr>
<tr>
<td>MobileNetV2</td>
<td>93.37%</td>
<td>2.22</td>
<td>16.08</td>
<td>1.70</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>97.66%</td>
<td>2.28</td>
<td>25.32</td>
<td>5.7</td>
<td></td>
</tr>
</tbody>
</table>

In terms of detection accuracy, the Swin-Transformer achieves the best performance, while the number of parameters and computational complexity are also higher, which are 12 times and 4 times of ours respectively. The MobileNetV2 baseline achieves the fastest detection speed and lowest computational complexity at the expense of 4.29% detection accuracy compared with our backbone. As a whole, our improved hybrid encoder achieves a good balance between the detection accuracy and the computational cost.

4.3.3 Comparative experiment of different types of positioning box

Experiments are conducted to compare the positioning accuracy of different types of box for the part with large aspect ratio. As shown in Figure 15, the TR _Corner part in MONA, is chosen to demonstrate the detection results of the rotated box and horizontal box. Figure 15(a) is the image taken by the camera, and Figure 15(b) is the image obtained according to the transformation relation between the standard template and in-site image. The experimental results show that the horizontal detection box contains a lot of background information, while the rotated box can effectively fit the parts, improving the accuracy of subsequent object matching.

![Detection results of different types of positioning boxes](image)

Figure 15 Detection results of different types of positioning boxes

4.4 Missing and wrong assembly errors inspection

In this section, the CNN-Transformer based assembly inspection method is used to carry out the missing and wrong assembly inspection experiment, and the workbench is shown in Figure 16. For each assembly step of MONA, sample images of qualified assembly and unqualified assembly are collected respectively, and the
assembly states in these images are recorded. A total of 150 sample images are collected, including qualified, missing and wrong assembly, and the number of each assembly sample is 50. The types of wrong assembly consist three common cases, part type error, angle error and position error. The precision, recall and F1 score are used in this paper as evaluation indexes for assembly fault inspection.

![Assembly workbench](image)

**Figure 16** Assembly workbench

Based on the positioning box matching algorithm, the IoU value between the detected box and the real positioning box is calculated. When the IoU result is less than the preset threshold, it is considered that the current step has a missing or wrong assembly error. After testing all samples, the final precision, recall and F1 score of three types of assembly samples detected by the proposed inspection method are calculated respectively, and the result is shown in **Table 5**. The CNN-Transformer based assembly inspection method proposed in this paper has achieved the F1 score of 90.91% for qualified assembly, 97.09% for missing assembly and 93.88% for wrong assembly. The inspection result of some samples is shown in **Figure 17**. The dashed line box is the real position of the part, and the solid line part is the bounding box detected by the proposed part detection model. Experimental results show that the proposed inspection method can effectively detect the assembly fault for each assembly step.

![Assembly quality inspection samples](image)

**Figure 17** Assembly quality inspection samples.
Table 5 Assembly quality inspection result

<table>
<thead>
<tr>
<th>Assembly type</th>
<th>Precision%</th>
<th>Recall%</th>
<th>F1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualified</td>
<td>91.84</td>
<td>90.00</td>
<td>90.91</td>
</tr>
<tr>
<td>Missing</td>
<td>94.34</td>
<td>100</td>
<td>97.09</td>
</tr>
<tr>
<td>Wrong</td>
<td>95.83</td>
<td>92.00</td>
<td>93.88</td>
</tr>
</tbody>
</table>

5 Conclusion

In this paper, an online inspection method based on deep learning and positioning box matching is proposed to inspect the assembly error. A novel hybrid neural network named CNN-Transformer is designed to integrate key point detection and part detection. The detected key points are used to solve the transformation relation between the in-site image and the template image. Based on this, the detected positioning box is matched with the standard positioning boxes to judge whether there are missing or wrong assembly problems. To verify the CNN-Transformer detection model, key point and part detection experiments are carried out, and the results are superior to other mainstream algorithms. At the same time, the CNN-Transformer-based method is used to inspect missing and wrong assembly errors. The results show that the average F1 score for missing and wrong assembly error identification reaches 93.96%.

In the future, the proposed assembly inspection algorithm will be deployed in embedded devices. Meanwhile, to reduce the burden of datasets preparation, unsupervised or weakly supervised detection methods based on CNN and Transformer structure will be explored.

Declarations

a. **Funding**

This research was supported by Defense Fundamental Research Foundation of China (JCKY2021203B072).

b. **Competing interests**

The authors have no competing interests to declare that are relevant to the content of this article.

c. **Data availability**

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

d. **Code availability**

Not applicable.
References


