IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 March 2023, accepted 21 April 2023, date of publication 26 April 2023, date of current version 2 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3270807

RESEARCH ARTICLE

ViT-SAPS: Detail-Aware Transformer for Mechanical Assembly Semantic Segmentation

HAITAO DONG[1], CHENGJUN CHEN[2], JINLEI WANG[2], FEIXIANG SHEN[2], AND YONG PANG[1,2]

[1]School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266520, China
[2]School of Mechanical and Automotive Engineering, Qingdao University of Technology, Qingdao 266520, China

Corresponding author: Chengjun Chen (chencj@qut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 52175471.

ABSTRACT Semantic segmentation of mechanical assembly images provides an effective way to monitor the assembly process and improve the product quality. Compared with other deep learning models, Transformer has advantages in modeling global context, and it has been widely applied in various computer vision tasks including semantic segmentation. However, Transformer pays the same granularity of attention on all the regions of an image, so it has some difficulty to be applied to the semantic segmentation of mechanical assembly images, in which mechanical parts have large size differences and the information quantity distribution is uneven. This paper proposes a novel Transformer-based model called Vision Transformer with Self-Adaptive Patch Size (ViT-SAPS). ViT-SAPS can perceive the detail information in an image and pays finer-grained attention on the regions where the detail information locates, thus meeting the requirements of mechanical assembly semantic segmentation. Specifically, a self-adaptive patch splitting algorithm is proposed to split an image into patches of various sizes. The more detail information an image region has, the smaller patches it is split into. Further, to handle these unfixed-size patches, a position encoding scheme and a non-uniform bilinear interpolation algorithm used after sequence decoding are proposed. Experimental results show that ViT-SAPS has stronger detail segmentation ability than the model with fixed patch size, and achieves an impressive locality-globality trade-off. This study not only provides a practical method for mechanical assembly semantic segmentation, but also has much value for the application of vision Transformers in other fields. The code is available at: https://github.com/QDLGARIM/ViT-SAPS.

INDEX TERMS Deep learning, vision Transformer, mechanical assembly monitoring, semantic segmentation.

I. INTRODUCTION

At present, the manufacturing industry has entered the era of mass customization. Mass customized production needs to constantly change product types according to the needs of different customers. This production mode with changeable product types makes the product assembly line constantly reorganized. This increases the assembly difficulty, and workers are prone to errors such as assembly procedure errors, missing assembly, and wrong assembly [1], [2], [3]. If these errors are not detected in time, it will directly affect the product quality and assembly efficiency. Therefore, effectively monitoring the assembly process and finding errors in time has become an urgent problem in the modern manufacturing industry. The traditional way to monitor the assembly process mainly relies on workers' visual comparison between assembly drawings and assembly products. This method is time-consuming and requires workers to have a high level of professional knowledge.

In recent years, image semantic segmentation has developed rapidly, and it can be applied to the field of mechanical assembly monitoring to replace the aforementioned manual labor and realize automatic monitoring. Specifically, RGB imaging sensors and depth imaging sensors are deployed on the assembly line to collect mechanical assembly images, and then semantic segmentation technology can be utilized to

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

segment these images. By analyzing the segmentation maps, the assembled parts can be identified, and the errors of missing assembly, wrong assembly, assembly procedure error, etc. can be monitored more precisely and efficiently than the traditional approach [2]. Thus, this method can improve the quality and efficiency of mechanical product assembly while avoiding the phenomenon of rework and reducing production costs.

At present, semantic segmentation based on deep learning is the mainstream semantic segmentation technology, which has been successfully applied to many fields, including virtual reality, autonomous driving, medical image analysis, etc. [4], [5], [6], [7]. However, applying semantic segmentation to mechanical assembly monitoring has some difficulties: (1) Mechanical assembly images have insufficient color and texture information, which increases the difficulty of mechanical assembly segmentation. (2) Information quantity distribution in an assembly image can be extremely uneven. Some regions of an assembly image may include a large number of tiny mechanical parts, whereas other regions may include mainly large mechanical parts or even meaningless backgrounds.

The convolution neural network (CNN) is a deep learning model mainly used for computer vision tasks. It was originally designed to imitate the animal optic nerve system and once occupied a dominant position in computer vision tasks such as image classification, object detection, and semantic segmentation [8]. CNN has already been applied to mechanical assembly monitoring. For example, Chen et al. [2] proposed an improved 3D convolutional neural network model (3D CNN) to detect missing assembly actions, and they used a fully convolutional network (FCN) to segment and identify different mechanical parts from complex assembly products to check whether there are missing or misplaced mechanical parts in the assembly process. Chen et al. [3] used YOLOv3 to locate and identify the assembly tools and identify the assembly behaviors of workers to prevent assembly quality degradation caused by the lack of key operation steps and misoperation. However, CNN has a limited receptive field, and it is not good at learning and modeling long-range dependency information in images [6], [7], [9], [10]. This drawback severely limits the application of CNN in mechanical assembly semantic segmentation. Since the mechanical assembly is a complex structure with many mechanical parts being closely connected and working together, the long-range dependency information among these parts needs to be considered in high-precision semantic segmentation of mechanical assembly.

In recent years, Transformer [11] has become a prevalent architecture in the field of natural language processing (NLP) attributed to its self-attention mechanism. Since 2020, Transformer has been successfully applied to computer vision tasks, including image classification, object detection, and semantic segmentation [8]. Such vision Transformer has a receptive field of the whole image, and it can leverage the global information of the image to overcome the disadvantage

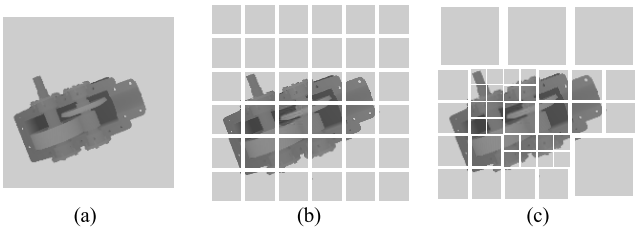


FIGURE 1. Patch splitting mechanisms in vision Transformers. (a) The original image; (b) The fixed-size patch splitting mechanism of the existing vision Transformers; (c) The adaptive patch splitting mechanism proposed in this paper.

of the limited receptive field of CNN [6], [7], [9], [10]. Thus, vision Transformer provides a new thinking for mechanical assembly semantic segmentation. However, there are still problems in applying the existing vision Transformers to mechanical assembly semantic segmentation directly. As shown in Fig. 1 (b), in the existing vision Transformers, an image is first split into a series of fixed-size (e.g., 16×16) image patches. Supposing the Transformer uses a constant latent vector size of D through all its layers, each patch is flattened and then mapped to D dimensions through a trainable linear projection. Therefore, an image is converted into a sequence of D-dimensional vectors, and then it can be processed by a Transformer. Since the patches have the same size, the Transformer pays the same granularity of attention on each region of the image. It cannot perceive the detail information inside a patch and pay special attention on it. If some detail information in the input image falls into a patch, it will be flattened into a vector, which may cause the loss of local context information. Therefore, although the existing vision Transformers have good global attention, it is still difficult for these architectures to deal with mechanical assembly images in which the information quantity distribution is extremely uneven.

To overcome the aforementioned limitation of the existing vision Transformers, this paper proposes a novel vision Transformer model called Vision Transformer with Self-Adaptive Patch Size (ViT-SAPS), which can pay finer-grained attention on image regions with more detail information. Compared with the existing vision Transformers, ViT-SAPS can split image patches adaptively according to the amount of detail information. If a patch has sufficient detail information, then it will be further split into smaller patches; otherwise, it will not be further split. As a result, the more detail information an image region has, the more attention is paid on this region to fully extract the internal information of an image. Fig. 1 shows the patch split results of a mechanical assembly image using the existing vision Transformers and ViT-SAPS. In the regions where tiny mechanical parts concentrate, ViT-SAPS has finer patch split granularity than the existing vision Transformers. In the regions where there are few tiny mechanical parts, ViT-SAPS has coarser patch split granularity than the existing vision Transformers. Therefore, ViT-SAPS can provide an effective

approach for semantic segmentation of mechanical assembly images with uneven information distribution.

Specifically, the contributions of the ViT-SAPS proposed in this paper include: (1) A self-adaptive image patch splitting algorithm for Transformers is proposed, which can adaptively split image patches according to the detail information distribution of the image. (2) Since ViT-SAPS use unfixed-size patches, and the positions of these patches cannot be effectively expressed by regular position encoding schemes, a special position encoding scheme is proposed to address this problem. (3) For unfixed-size patches, a non-uniform bilinear interpolation algorithm used after sequence decoding is also proposed.

The rest of this paper is organized as follows. Section II presents an overview of current research. Section III outlines the research process and describes the self-adaptive patch splitting algorithm, the position encoding scheme for unfixed-size patches, and the non-uniform bilinear interpolation algorithm in detail. Section IV presents experiments conducted using the proposed method, including the datasets, experiment settings, and the performance of ViT-SAPS. Section V presents our conclusions and future work.

II. RELATED WORK

CNN-based semantic segmentation has been researched extensively. In 2015, Long et al. [12] proposed FCN, which laid the foundation for semantic segmentation research based on deep learning. FCN realizes full convolution so that the network can accept the input of any size, and the output of the network is a spatial image rather than a class score. Although FCN has made great progress compared with traditional semantic segmentation methods based on artificial feature extraction, FCN still has some limitations. When FCN outputs the final segmentation map, it uses deconvolution for 8 × upsampling, which makes FCN insensitive to image details and leads to rough semantic segmentation results. When FCN performs pixel-by-pixel segmentation, it does not consider the relationship among pixels and lacks spatial consistency. To overcome these shortcomings, researchers have proposed a series of neural network models for semantic segmentation based on FCN, such as CRF-RNN [13] based on conditional random field (CRF), DPN [14] based on Markov random field (MRF), U-Net [15] and SegNet [16] using multi-level feature fusion encoder-decoder structure, etc. These models have continuously improved the segmentation accuracy and efficiency.

However, these FCN-based models have a limited receptive field, so they are not good at extracting global context information. To address this problem, a lot of work has been conducted. For example, DeepLab [17] and Dilation [18] use dilated convolution to expand the receptive field. PSPNet [19] and DeepLabV2 [20] use pyramid pooling to aggregate the context information of different regions to improve the ability to obtain global information. PSANet [21] proposes a point-wise spatial attention module to dynamically capture long-range context; DANet [22] proposes dual attention composed of position attention and channel attention to capture the global feature dependency in the spatial and channel dimensions respectively. In GCAU-Net [23], a global attention module with self-attention mechanism is proposed to learn the long-range dependencies of channel and position, and a feature aggregation module is designed to fuse the global context features with the low-level and high-level of features generated by a CNN-based encoder. The above models have achieved tremendous progress in modeling global context. However, these models are still constructed based on FCN, and the spatial resolution of the input image is still downsampled by the encoder, so some global contextual information will still be lost.

In 2020, Transformer was extended to the field of computer vision [24]. Vision Transformers split the input image into a series of fixed-size image patches, thus representing images as sequences and making image processing tasks similar to natural language processing tasks. The Transformer encoder has a global receptive field. It does not perform downsampling, and has better modeling ability than CNN [25]. So, it provides a new idea for semantic segmentation. In the past two years, the research on Transformer-based semantic segmentation has made many achievements [8]. For example, SETR [9] is a Transformer-based semantic segmentation model, which uses Vision Transformer (ViT) [26] as the encoder, and utilizes three different decoders (Naive, PUP and MLA) to perform semantic segmentation. Segmenter [10] is also a Transformer-based semantic segmentation model, and it differs from SETR in that: (1) In addition to ViT, the encoder of Segmenter also uses DeiT [27], which has lower requirements for the scale of the training set; (2) The decoder adopts a pure Transformer architecture, called mask Transformer. TransUNet [28] is a special model for medical image segmentation, which combines the advantages of Transformer and U-Net. In TransUNet, the input of the Transformer is the patches split from the feature map output from a CNN rather than the original image. This is conducive to the extraction of global context information. Then, the decoder upsamples the output features of the encoder and then combines them with high-resolution CNN feature maps to achieve an accurate target location. These early Transformer-based semantic segmentation models have a disadvantage, i.e., their Transformer backbones work based on the patch splitting mechanism shown in Fig. 1 (b), and thus they show poor locality in practice. In fact, these backbones perform well in image classification tasks but perform poorly in tasks requiring dense prediction at the pixel level, such as object detection and semantic segmentation [29]. Therefore, the performance of these early Transformer-based semantic segmentation models is not satisfactory.

To overcome this disadvantage, a plethora of studies have been conducted on the locality of vision Transformers. In some research works [5], [28], CNNs are used to enhance the locality of vision Transformers. In Focal Transformer [30], a new self-attention mechanism is proposed. When calculating the attention between a query patch

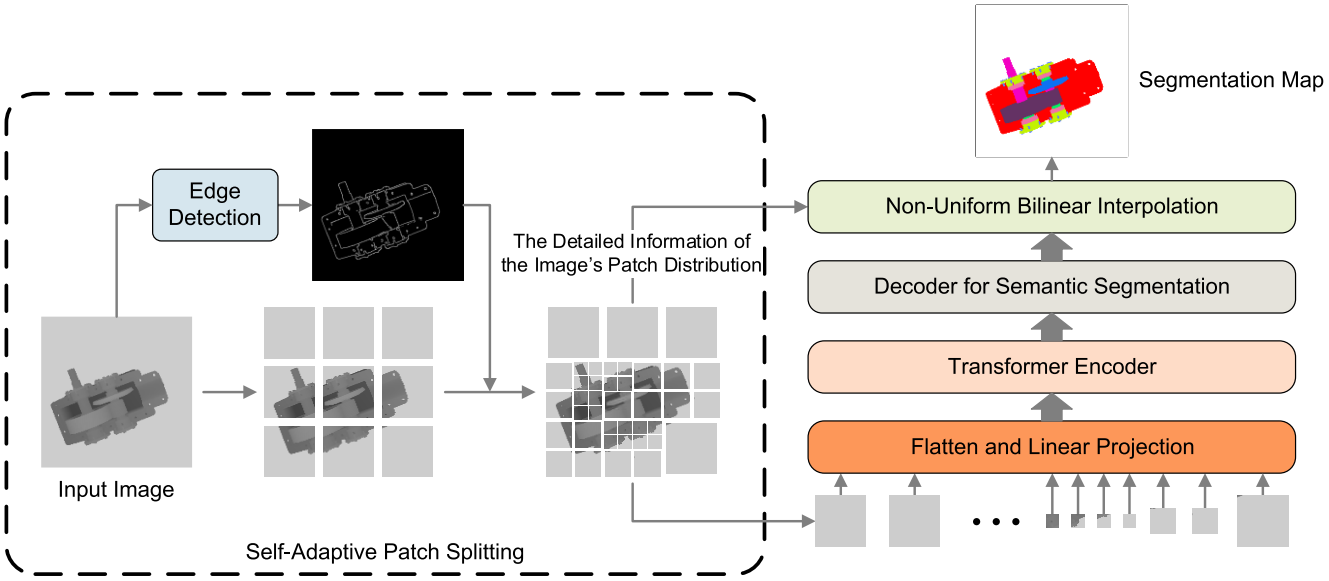


FIGURE 2. The architecture of the proposed ViT-SAPS.

and other regions of the image, fine-grained self-attention is performed in the closest surrounding regions whereas coarse-grained self-attention is performed in the regions far away. VOLO [31] is also a new self-attention mechanism, which encodes fine-level local contexts into image tokens in a way similar to patch-wise dynamic convolution. The Transformer-based semantic segmentation model LSNet [32] proposes a novel context formation fusion method based on the cross attention structure, which fuses the global attention features and low-level spatial features, and thus improves the model's accuracy in segmenting complex targets. D-Former [6] proposes a local scope multi-head self-attention mechanism to provide locality, which performs self-attention only within each local unit in the image. Tokens-to-Token ViT [33] aggregates several neighboring patches into one so that the aggregated patch has local information and solves the problem of the poor locality of ViT. TNT [34] uses a two-level Transformer: the outer Transformer is responsible for global attention modeling, and the inner Transformer is responsible for local attention modeling, thus unifying global attention and local attention. PVT [29] is a general-purpose Transformer backbone for computer vision tasks such as image classification, object detection, semantic segmentation, etc. PVT adopts a hierarchical architecture, and different patch sizes are used at different stages, so it can consider both locality and globality. Similar to PVT, Swin Transformer [35], [36] is also a hierarchical general-purpose Transformer backbone, which utilizes a shifted window along the spatial dimension to reduce the model's computational complexity. Swin-Unet [7] is a pure Transformer-based semantic segmentation model designed using the backbone of Swin Transformer, and it is a U-shape model similar to U-Net. Swin-Unet performs better than CNN models and CNN-Transformer hybrid models in medical image segmentation. Some other vision Transformers also adopt hierarchical architecture to improve locality, such as SegFormer [37], Dynamic Transformer [38], Twins [39], Multi-Scale Vision Longformer [40], etc. Although these models have greatly improved the locality of vision Transformers, they still pay the same granularity of attention on all the regions of the input image and do not fully consider the characteristics of mechanical assembly images. Therefore, it is necessary to develop a vision Transformer suitable for mechanical assembly semantic segmentation.

III. METHOD

A. OVERALL ARCHITECTURE

The architecture of the ViT-SAPS model proposed in this study is shown in Fig. 2. The input image is first split into several (e.g. 3×3) large patches. Then, the image is further split according to the edge detection result. If an aforementioned large patch contains sufficient edge pixels, it is considered to contain abundant high-frequency components, and thus it has much detail information and will be further split into 2×2 smaller sub-patches; otherwise, if a patch contains few edge pixels, it is considered to have little detail information and it will not be further split. In this way, ViT-SAPS can adaptively split an image into unfixed-size patches according to the distribution of detail information in the image. For image regions where the detail information locates, ViT-SAPS gives finer-grained attention to fully extract the key information. For image regions that contain little detail information, ViT-SAPS use large patches, thus effectively reducing the sequence length of the image and the computational complexity of the model without affecting the segmentation performance.

Next, regardless of the patch size, all the patches will be flattened and linearly transformed into vectors of the same

dimension. Then, these vectors are successively processed by the Transformer encoder and semantic segmentation decoder to obtain a patch-level class score. In this process, we adopt position encoding to retain positional information of the patches. In ViT-SAPS, the characteristic of unfixed-size patches makes both the number and distribution of patches differ among the images. As a result, the fixed learnable position encoding of the existing vision Transformers cannot be applied to this situation. Instead, we propose a position encoding scheme specially for ViT-SAPS.

Finally, the patch-level class score is subjected to bilinear interpolation to obtain the final segmentation map. Since the patches are distributed irregularly in each image, the bilinear interpolation algorithm of the existing vision Transformers is unsuitable for ViT-SAPS. To address this problem, we propose a non-uniform bilinear interpolation algorithm. The non-uniform bilinear interpolation process is guided by the detailed information of the input image's patch distribution.

The self-adaptive patch splitting algorithm, the position encoding scheme, and the non-uniform bilinear interpolation algorithm proposed in this paper will be discussed in detail in the following subsections.

B. SELF-ADAPTIVE PATCH SPLITTING ALGORITHM

The purpose of self-adaptive patch splitting is to split the image regions with more detail information into smaller patches and split the image regions with fewer details into larger patches. Hence, this study proposes a self-adaptive patch splitting algorithm (Fig. 2). Considering an image $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$, the algorithm flow is described as follows.

1) For an image in the dataset, its height H and width W are factorized as $H = h \times 2^m$ and $W = w \times 2^m$, where at least one of h and w is 1 or an odd prime, and m is a positive integer. To facilitate the execution of the subsequent steps, 2^m should be a relatively large value, such as 128 or even larger. If this condition is not met, padding needs to be employed on the image in advance.
2) According to the factorization results, the image is firstly split into h × w large patches sized $2^m \times 2^m$.
3) Edge detection is performed on the image to obtain an edge-detected image.
4) According to the edge-detected image, the total number of edge pixels g in each large patch described in step 2) is examined. If g is not smaller than a preset threshold T_{split}, then this patch is considered to contain much detail information. Therefore, this patch is split into 2 × 2 sub-patches sized $2^{m-1} \times 2^{m-1}$. Then, the above splitting process is recursively performed until the patch size is reduced to a preset lower threshold $T_{patch} \times T_{patch}$. T_{patch} should be an integer power of 2. If $g < T_{split}$, no further split will be continued.

After the above process, the image is split into a sequence of patches $\mathbf{x} = [x_1, \ldots, x_N]$ of different sizes. The maximum and minimum patch sizes are $2^m \times 2^m$ and $T_{patch} \times T_{patch}$ respectively. The threshold T_{split} may have effect on the sequence length N. The smaller T_{split} is set, the more likely the image is to be split into smaller patches, and thus the sequence length is larger. Meanwhile, different images may have different sequence lengths N. Similar to the processing of the existing vision Transformers, each patch is flattened into a one-dimensional vector and then transformed into a patch embedding $\mathbf{x}_0 = [x_{0,1}, \ldots, x_{0,N}] \in \mathbb{R}^{N\times D}$ of length D through a linear projection. For the existing vision Transformers, because each patch has the same size, only one learnable linear projection is required. However, for ViT-SAPS, because the patches are of different sizes, each group of patches with the same size requires an independent learnable linear projection. Meanwhile, since the patch splitting operation of each image is different, after the patch splitting, some detailed information about the patch distribution of each image, including each patch's size and its specific position in the original image, needs to be transmitted to the interpolation module following the decoder to restore the segmentation map effectively.

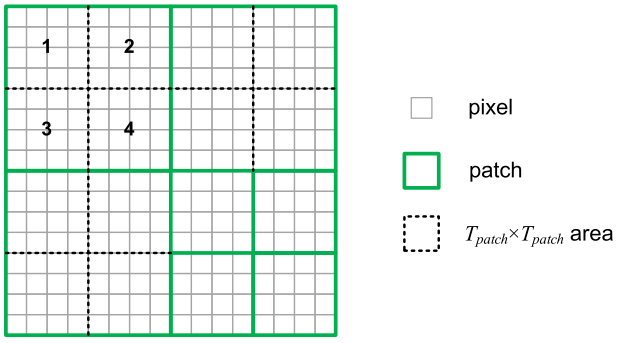


FIGURE 3. The proposed position encoding scheme. This schematic diagram is shown for an image of 16 × 16 pixels. The squares with gray borders represent pixels in the image. The squares with green borders represent the patches. In this schematic diagram, $T_{patch} = 4$, so there are 16 $T_{patch} \times T_{patch}$ areas in total. Each $T_{patch} \times T_{patch}$ area has a learnable position encoding like ViT. Then, the position encoding of the 8 × 8 patch located in the upper left corner of the image is equal to the average of the position encodings of the four $T_{patch} \times T_{patch}$ areas of 1, 2, 3, and 4.

C. POSITION ENCODING SCHEME

In the existing vision Transformers, the patch distribution of each input image is consistent. This indicates that each token in the patch sequence has a fixed position in the original image. Therefore, a unified learnable position encoding can be used for all the images to represent the position information of each patch in the original image. However, this scheme is not suitable for ViT-SAPS. This is because, in ViT-SAPS, the lengths of the patch sequences are inconsistent, the sizes of the patches are different, and the positions of the patches are not fixed. Hence, it is difficult for ViT-SAPS to use fixed patch position encoding like the existing vision Transformers. To address this problem, this study designs a position encoding scheme for ViT-SAPS, which is described as follows.

First, each image is split into a series of small areas sized $T_{patch} \times T_{patch}$, where $T_{patch} \times T_{patch}$ is the minimum patch

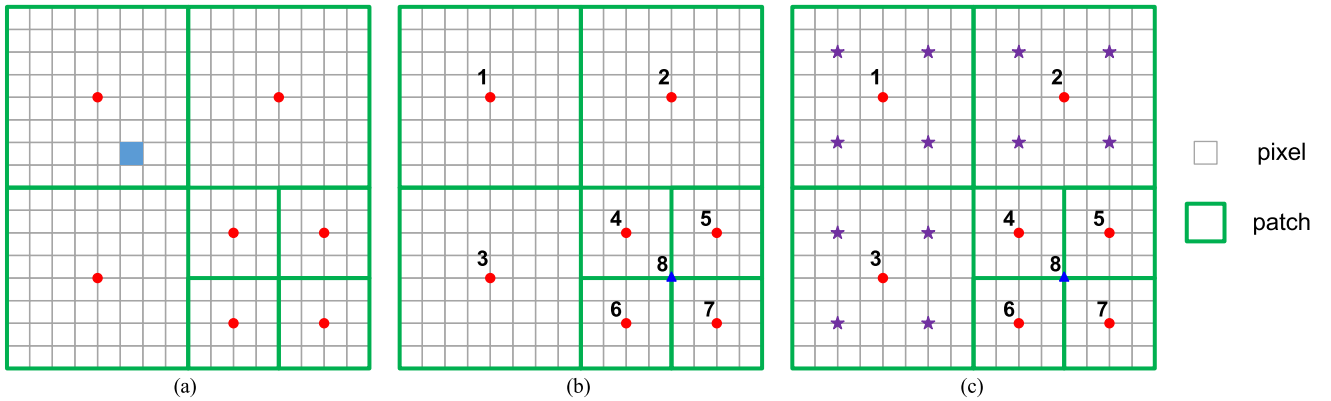


FIGURE 4. The proposed non-uniform bilinear interpolation algorithm. In this schematic diagram, (a), (b), and (c) represent an image of 16 × 16 pixels. The squares with gray fine borders represent pixels in the image. The squares with green bold borders represent the patches. The red dots represent points with known class scores, i.e., the center of each patch. The centers of the pixels are the points to be interpolated. Because the pixel marked in blue in Fig. 4(a) is not in the rectangular area determined by any four known points, according to the principle of bilinear interpolation, the class score of the blue pixel cannot be directly determined by bilinear interpolation. The value of the dark blue triangle 8 in Fig. 4(b) is determined by performing bilinear interpolation on the four points 4, 5, 6, and 7. The values of the 12 purple stars in Fig. 4(c) are determined by performing bilinear interpolation on the four points 1, 2, 3, and 8. These 12 points, plus points 4, 5, 6, and 7, whose values are originally known, can be used to determine the value of each pixel in the image using the ordinary bilinear interpolation method.

size, so that each patch in the image contains several such $T_{patch} \times T_{patch}$ areas. Then, similar to ViT, a learnable position encoding is arranged for each $T_{patch} \times T_{patch}$ area. The position encoding of each patch takes the mean value of the position encodings of all the $T_{patch} \times T_{patch}$ areas it contains (Fig. 3). Such position encoding scheme can correctly reflect each patch's position in the original image, although the distribution of patches in each image can be totally different.

In this way, the position encoding $\mathbf{pos} = [pos_1, \ldots, pos_N] \in \mathbb{R}^{N \times D}$ is obtained and added to the patch embedding, thereby obtaining the input sequence of the Transformer:

$$\mathbf{z}_0 = \mathbf{x}_0 + \mathbf{pos} \tag{1}$$

D. SEQUENCE ENCODING AND DECODING

In ViT-SAPS, the encoding and decoding process of the sequence is the same as that of the existing Transformer-based semantic segmentation models. The mapping of the l-th layer in the Transformer encoder is denoted as $\text{Encoder}_l(\,)$, and the output sequence is denoted as $\mathbf{z}_l = [z_{l,1}, \ldots, z_{l,N}] \in \mathbb{R}^{N \times D}$. Then,

$$\mathbf{z}_l = \text{Encoder}_l(\mathbf{z}_{l-1}), \quad l = 1, 2, \cdots, L \tag{2}$$

The input sequence $\mathbf{z}_0$ is processed by the Transformer encoder with a total of L layers. The final output sequence is $\mathbf{z}_L$. Subsequently, through the decoder, the sequence $\mathbf{z}_L$ is decoded into a patch-level class score $\mathbf{z}_{class} \in \mathbb{R}^{N \times K}$, where K is the number of classes. That is,

$$\mathbf{z}_{class} = \text{Decoder}(\mathbf{z}_L) \tag{3}$$

E. NON-UNIFORM BILINEAR INTERPOLATION ALGORITHM

In the existing Transformer-based semantic segmentation models, $\mathbf{z}_{class}$ is transformed into a patch-level segmentation map $\mathbf{s}_{patch} \in \mathbb{R}^{(H/P) \times (W/P) \times K}$, where P is the fixed patch size. Then, the final pixel-level segmentation map $\mathbf{s} \in \mathbb{R}^{H \times W \times K}$ is obtained by performing bilinear interpolation on $\mathbf{s}_{patch}$. In ViT-SAPS, since the patches are non-uniformly distributed on the original image, after $\mathbf{z}_{class}$ is obtained, the class score of each pixel cannot be directly obtained by bilinear interpolation, as shown in Fig. 4(a).

Considering the above problems, this study proposes a bilinear interpolation algorithm for non-uniformly distributed patches. With the help of the previously generated detailed information about patch distribution (Fig. 2), the proposed algorithm uses the patch-level class score $\mathbf{z}_{class}$ in the non-uniformly distributed patches to complete the interpolation computation of the class score of each pixel. According to this method, in the final segmentation map $\mathbf{s}$, the smaller the patches, the denser the class scores obtained by deep learning rather than by interpolating, and thus the higher the interpolation precision. Hence, the segmentation precision is also higher. On the contrary, the larger the patches, the lower the interpolation precision. Thus, the segmentation precision is also lower. This is consistent with the goal of this study to improve the segmentation precision of vision Transformers at image regions with detail information. The details of the algorithm are described as follows, and the flowchart is shown in Fig. 5.

1) The size of the minimum patch in the image being processed is denoted as $p_{\min} \times p_{\min}$, and the size of the maximum patch is denoted as $p_{\max} \times p_{\max}$. Let $i = p_{\min}$. Each $i \times i$ patch in the image is obtained by splitting a $2i \times 2i$ patch. Then, bilinear interpolation is exploited to determine the values of the center points of these $2i \times 2i$ patches. For example, in Fig. 4(b), the value of point 8 can be obtained by performing bilinear interpolation on points 4, 5, 6, and 7.

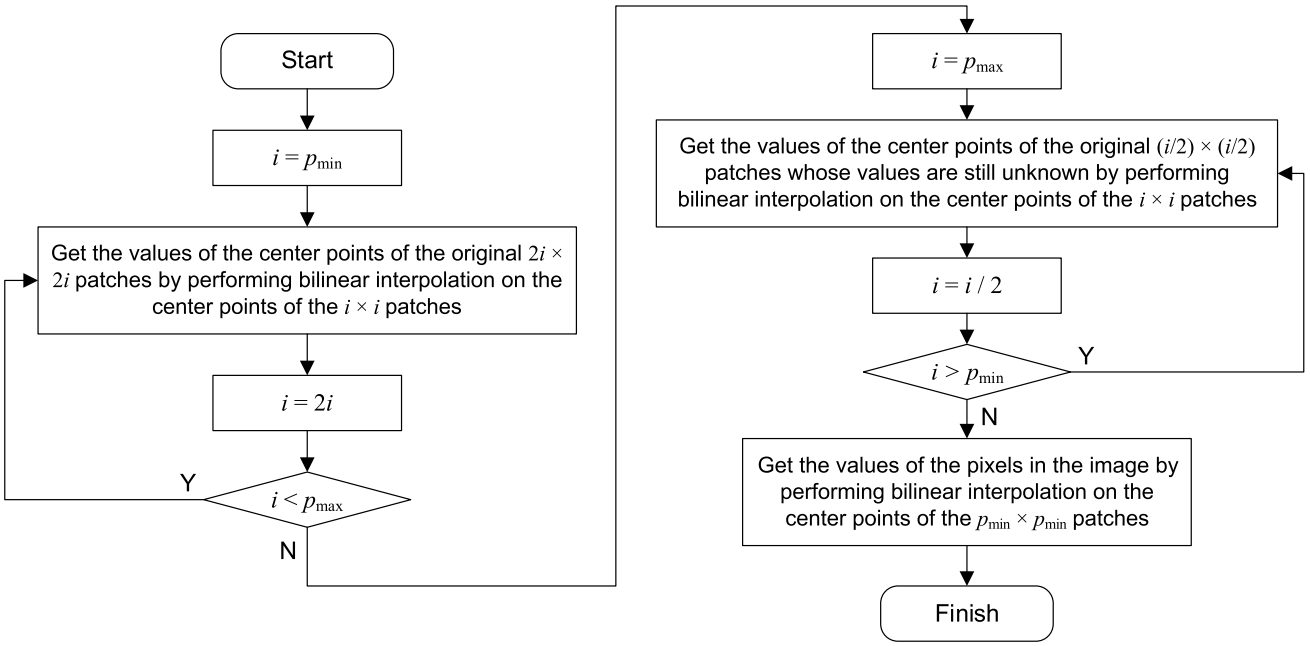


FIGURE 5. The flowchart of the proposed non-uniform bilinear interpolation algorithm.

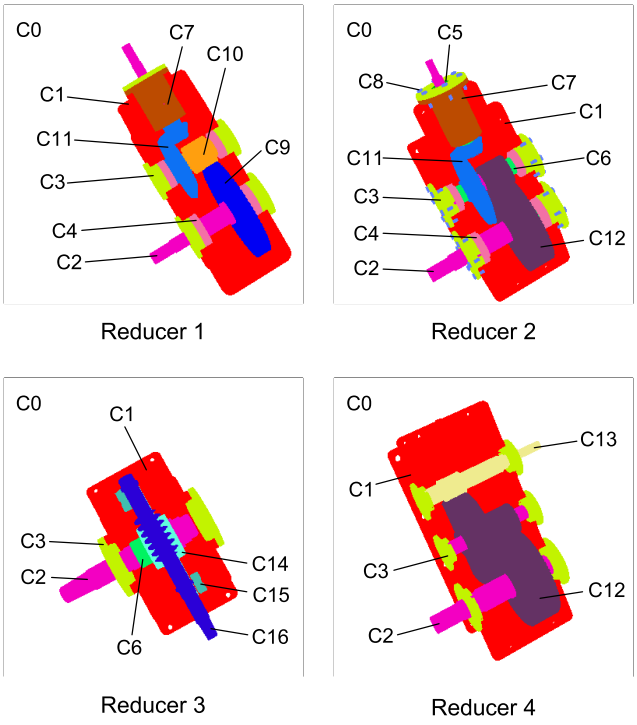
2) Let $i = 2i$. If $i < p_{\max}$, the interpolation process in step 1) is repeated, i.e., the center point of each $2i \times 2i$ patch is obtained by performing bilinear interpolation on the values of the center points of $i \times i$ patches in the image; otherwise, go to step 3). After the interpolation in steps 1) and 2), the values of the center points of $p_{\max} \times p_{\max}$ patches in the image are known. This is a necessary condition for the interpolation in step 3).
3) Let $i = p_{\max}$. The center points of $i \times i$ patches in the image are used to perform bilinear interpolation to determine the values of the center points of $(i/2) \times (i/2)$ patches whose values are still unknown. For example, in Fig. 4(c), it is assumed that the image is split into patches sized 4×4. Hence, there are 16 such patches in total. In these patches, the values of the center points 4, 5, 6, and 7 are known. The remaining 12 center points can be determined by performing bilinear interpolation on points 1, 2, 3, and 8.
4) Let $i = i/2$. If $i > p_{\min}$, the interpolation process in step 3) is repeated, i.e., the center point of each $(i/2) \times (i/2)$ patch whose value is still unknown is obtained by performing bilinear interpolation on the center points of $i \times i$ patches in the image; otherwise, go to step 5). After the interpolation in steps 3) and 4), it is assumed that the image is split into patches sized $p_{\min} \times p_{\min}$. Then, the values of the center points of such patches are already known. This is a necessary condition for the interpolation of step 5).
5) The ordinary bilinear interpolation is performed on the center points of the above-mentioned patches sized $p_{\min} \times p_{\min}$ to obtain the class scores of all the pixels in the image. In this way, the complete segmentation map $\mathbf{s} \in \mathbb{R}^{H \times W \times K}$ is obtained.

IV. EXPERIMENTS

A. DATASETS

Currently, there is still a lack of public datasets in the field of mechanical assembly semantic segmentation. Therefore, this study conducts experiments on a private dataset we built. In the practical production process, since the color and texture information of the mechanical assembly is lacking, it is not suitable to use color images for mechanical assembly semantic segmentation. The pixels in the depth image indicate the distance between the actual object and the sensor. Hence, the depth image can describe the three-dimensional information of the object. Besides, compared with color images, depth images are not disturbed by environmental factors such as illumination, chromaticity, and shadows. Thus, depth images are suitable for the semantic segmentation of mechanical assemblies. Depth images can be acquired by depth cameras. However, there are various mechanical parts in common mechanical assemblies, and some are tiny in size, which makes it difficult and laborious to manually label real depth images. Therefore, massive training samples cannot be obtained easily. Considering this, our dataset adopts a computer synthesis method to generate mechanical assembly depth images and the corresponding labels.

Our data samples contain depth images and their labels of four reducer products at different assembly stages. There are 3004 data samples in total, with image size of 384×384 pixels. The appearances of the parts in each mechanical product are shown in Fig. 6. The details of the parts included in



Index	Mechanical part types
C0	Background
C1	Cabinet
C2	Shaft
C3	Bearing end-shield
C4	Bearing
C5	Bearing sleeve (Type 1)
C6	Bearing sleeve (Type 2)
C7	Sleeve
C8	Bolt
C9	Cylindrical spur gear
C10	Cylindrical spur gear shaft
C11	Bevel gear
C12	Cylindrical helical gear
C13	Cylindrical helical gear shaft
C14	Worm gear
C15	Oil scraper
C16	Worm

FIGURE 6. The appearances of the parts in each mechanical product.

each assembly stage of each product are presented in Table 1. The details of the number of samples in each assembly stage of each product are presented in Table 2.

Based on these data samples, two datasets are built for two different assembly monitoring tasks. The details of the datasets are as follows.

Dataset 1: All the 3004 data samples are first randomly shuffled and then divided at certain proportions to form the training set, the validation set and the test set, which contain 1804, 600, and 600 samples respectively. This dataset can be used at the situation where massive random sampling is performed to detect errors in the mechanical assembly process, so that the defective product rate can be obtained, and thereby the assembly quality can be determined.

Dataset 2: In this dataset, one certain assembly stage is selected from each of the four reducer products to form the test set, including Stage 3 of Reducer 1, Stage 5 of Reducer 2, Stage 1 of Reducer 3, and Stage 2 of Reducer 4. Consequently, there are 646 samples in the test set. The remaining 2358 samples are randomly shuffled, and then divided to form the training set and the validation set. There are 1768 and 590 samples in the training set and the validation set, respectively. This dataset can be used to monitor the assembly sequence of complicated products, i.e., to monitor the assembly process, and determine whether there are any mistakes in the assembly sequence.

B. EXPERIMENT SETTINGS

We replace the fixed-size patch splitting mechanism in the DeiT-S [27] backbone with our self-adaptive patch splitting mechanism to form a novel vision Transformer backbone called DeiT-S-SAPS. Then, Mask Transformer [6] is taken as the decoder to form a complete ViT-SAPS semantic segmentation model. Due to the small size of our mechanical assembly semantic segmentation dataset, the DeiT-S backbone pre-trained on ImageNet-22K [41] is used in our model.

In the patch splitting process, the edge detection method should be carefully selected. We perform four popular edge detection operators [42], [43] on an image of Reducer 2, Stage 6. The results are presented in Fig. 7. In the edge-detected images generated by the Sobel, Scharr, and Laplacian operators, larger mechanical parts tend to have wider and clearer edges, while smaller mechanical parts tend to have finer and less clear edges. Using this type of edge-detected image in the patch splitting process is not conducive to the segmentation of small mechanical parts. It is desired that all the mechanical parts have edges of the same width and gray value. The Canny operator generates edges with a line width of 1 pixel and a unified gray value, and thus perfectly meets our requirements. Therefore, we adopt the Canny operator to conduct our experiments.

In the experiments, the ViT-SAPS model is evaluated under two situations: $T_{split} = 15$, $T_{patch} = 4$ and $T_{split} = 3$, $T_{patch} = 4$. Among them, the situation of $T_{split} = 15$ is adopted to compare the self-adaptive patch splitting mechanism with the traditional fixed-size patch splitting mechanism. When $T_{split} = 15$, in all the 3004 data samples, each image contains 592.8 patches on average. The traditional Transformer semantic segmentation model for comparison uses fixed size patches, and the size of each patch is 16×16. Thus, each

TABLE 1. The details of the parts included in each assembly stage of each mechanical product.

Mechanical products	Assembly stage	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
Reducer 1	1	√	√	√	√					√							
	2	√	√	√	√					√	√	√					
	3	√	√	√	√					√	√	√					
	4	√	√	√	√			√		√	√	√					
Reducer 2	1	√	√	√					√								
	2	√	√	√	√		√		√				√				
	3	√	√	√	√		√		√				√				
	4	√	√	√	√		√		√			√	√				
	5	√	√	√	√	√	√		√			√	√				
	6	√	√	√	√	√	√	√	√			√	√				
Reducer 3	1	√	√	√											√		
	2	√	√												√	√	
	3	√	√	√			√								√	√	
	4	√	√	√			√								√	√	√
Reducer 4	1	√	√	√									√				
	2	√	√	√									√				
	3	√	√	√									√				
	4	√	√	√									√	√			

In this table, the check marks represent that such parts are included.

TABLE 2. The details of the number of data samples in each assembly stage of each mechanical product.

Assembly stage	1	2	3	4	5	6	Total
Reducer 1	180	180	160	180	-	-	700
Reducer 2	180	180	180	180	180	180	1080
Reducer 3	180	180	180	180	-	-	720
Reducer 4	126	126	126	126	-	-	504

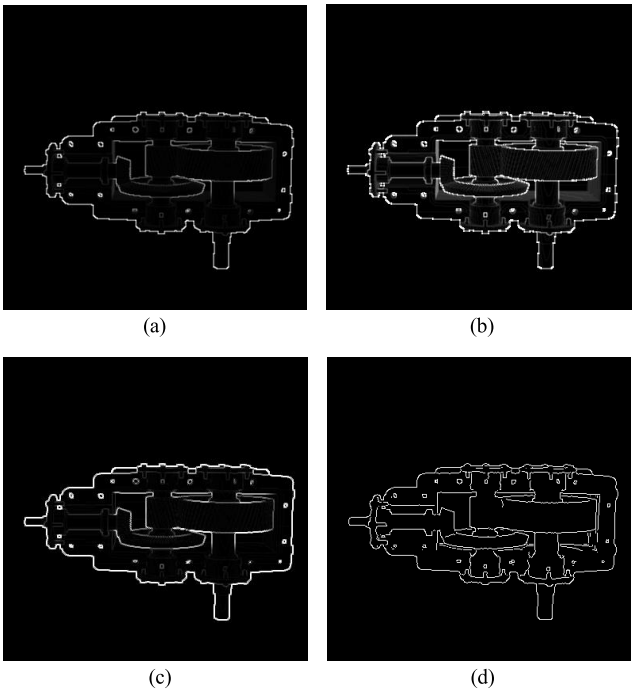


FIGURE 7. Edge detection results generated by 4 different operators. (a) Sobel; (b) Scharr; (c) Laplacian; (d) Canny.

image contains $(384 / 16)^2 = 576$ patches. Therefore, the two models have similar average input sequence lengths, and it is fair to compare the two models in this case. When $T_{split} = 3$, each image contains up to 1604.8 patches on average, which can provide more fine-grained semantic segmentation. This case is adopted to reflect the high performance of ViT-SAPS.

To achieve the fastest running speed, the encoder and decoder of ViT-SAPS are run on GPU similar to other Transformer-based semantic segmentation models. Since patch split and edge detection involve quite little parallel computation, these works are run on CPU rather than GPU. In non-uniform bilinear interpolation, there is also quite little parallel computation. Hence, this work is also performed on CPU rather than GPU.

C. RESULTS

We compare the performance of the proposed ViT-SAPS model with that of several classic semantic segmentation models and several recently proposed semantic segmentation models that perform excellently in other application fields on our Datasets 1 and 2. The results of segmentation intersection-over-unions (IoU) of 16 types of mechanical parts are presented in Table 3 and Table 4.

1) ANALYSIS OF THE SELF-ADAPTIVE PATCH SPLITTING MECHANISM'S VALIDITY

We demonstrate the validity of the proposed self-adaptive patch splitting mechanism by comparing ViT-SAPS ($T_{split} = 15$) and Segmenter, as shown in Table 3 and Table 4. These two models have the same encoder and decoder. The only difference between them is that ViT-SAPS ($T_{split} = 15$) uses the self-adaptive patch splitting mechanism, while Segmenter uses the traditional fixed-size patch splitting mechanism. It can be seen from Table 3 and Table 4 that ViT-SAPS ($T_{split} = 15$) achieves higher segmentation performance for all the mechanical parts except bearing sleeve 1 (C5) than Segmenter. The superiority of ViT-SAPS is especially significant in the segmentation of tiny parts such as bearing end-shield (C3), bearing sleeve 2 (C6), bolt (C8), and oil

TABLE 3. The segmentation IoU of the mechanical parts (Dataset 1).

Model	FCN	FCN	U-Net	PSPNet	GCAU-Net ($n = 2$)	Segmenter	Swin-Unet	ViT-SAPS ($T_{split} = 15$)	ViT-SAPS ($T_{split} = 3$)
Backbone	VGG16	VGG19	-	ResNet50	-	DeiT-S	Swin-T	DeiT-S-SAPS	DeiT-S-SAPS
Decoder	-	-	-	-	-	Mask Transformer	-	Mask Transformer	Mask Transformer
#Params.	18.6M	24.0M	31.0M	49.1M	20.0M	26.6M	41.4M	54.8M	54.8M
Patches on average	-	-	-	-	-	576	-	592.8	1604.8
C1	99.18	99.26	99.73	97.81	**99.77**	93.65	99.28	95.96	97.89
C2	97.11	97.55	99.03	96.41	**99.19**	88.10	97.42	92.35	95.56
C3	90.74	91.76	95.39	92.95	**97.28**	80.73	92.40	88.36	92.97
C4	81.01	82.01	87.38	92.90	**94.95**	80.40	86.46	85.46	89.29
C5	0	0	0	0	0	0	0	0	0
C6	77.74	81.53	88.20	**91.92**	88.68	72.21	86.61	78.53	84.09
C7	53.98	43.79	86.11	**96.49**	95.15	90.63	84.16	92.13	93.74
C8	52.11	50.49	82.90	57.20	**88.61**	3.72	67.63	44.23	69.37
C9	86.33	93.93	98.65	97.51	**99.43**	91.91	97.00	94.14	96.78
C10	88.83	96.41	98.22	95.89	**98.34**	91.01	96.42	91.76	94.53
C11	82.96	89.14	96.39	96.36	**98.78**	87.84	96.54	93.10	95.92
C12	91.91	96.49	**98.69**	98.62	95.05	95.22	98.76	96.85	98.14
C13	21.10	18.02	43.27	93.92	84.43	84.49	**95.45**	90.96	94.61
C14	71.65	95.62	98.63	93.06	**99.41**	87.23	97.59	91.24	94.83
C15	63.89	58.11	85.97	**87.06**	85.88	57.97	75.63	69.48	79.28
C16	25.58	43.83	87.13	91.98	90.03	87.47	58.90	90.88	**94.78**
mIoU	67.76	71.12	84.11	86.26	**88.44**	74.54	83.14	80.96	85.74

TABLE 4. The segmentation IoU of the mechanical parts (Dataset 2).

Model	FCN	FCN	U-Net	PSPNet	GCAU-Net ($n = 2$)	Segmenter	Swin-Unet	ViT-SAPS ($T_{split} = 15$)	ViT-SAPS ($T_{split} = 3$)
Backbone	VGG16	VGG19	-	ResNet50	-	DeiT-S	Swin-T	DeiT-S-SAPS	DeiT-S-SAPS
Decoder	-	-	-	-	-	Mask Transformer	-	Mask Transformer	Mask Transformer
#Params.	18.6M	24.0M	31.0M	49.1M	20.0M	26.6M	41.4M	54.8M	54.8M
Patches on average	-	-	-	-	-	576	-	592.8	1604.8
C1	97.40	97.51	**98.79**	96.74	98.38	90.22	98.62	93.88	96.73
C2	75.28	76.74	76.81	**79.06**	78.07	67.01	76.95	71.30	74.60
C3	92.44	92.57	**95.59**	90.96	94.66	76.33	92.56	82.97	87.86
C4	62.02	62.32	61.02	**71.25**	66.00	59.92	67.03	62.19	63.44
C5	0	0	0	0	0	0	0	0	0
C6	25.09	22.07	37.33	33.66	38.05	30.12	**38.54**	32.73	35.90
C7	0	0	0	0	0	0	0	0	0
C8	33.82	29.94	35.01	38.57	36.27	1.58	51.13	32.02	**53.53**
C9	75.23	66.53	53.88	71.37	51.16	90.80	**95.81**	91.79	94.48
C10	64.53	64.63	**96.19**	72.95	95.63	87.77	94.84	88.52	90.07
C11	79.09	71.41	70.10	81.66	69.04	84.85	**94.79**	89.82	92.55
C12	87.15	74.31	92.53	83.03	95.31	86.20	**95.73**	88.14	90.23
C13	0	0	0	0	0	-	0	-	-
C14	53.48	45.08	**98.68**	97.15	92.92	92.49	88.00	95.67	97.51
C15	0	0	0	0	0	0	0	0	0
C16	0	0	0	0	0	-	0	-	-
mIoU	46.60	43.95	51.00	51.03	50.97	54.81	55.87	59.22	**62.64**

scraper (C15). Bearing sleeve (Type 1) (C5) is a mechanical part installed inside the bearing end-shield (C3) of Reducer 2 (Fig. 5). This part is difficult to observe from outside the assembly, so all the models tested in this paper fail to recognize it. Note that when using the DeiT-S-SAPS backbone in the model, the number of model parameters is more than twice that of using the DeiT-S backbone. This is because in DeiT-S-SAPS, an independent learnable linear projection is required for each group of patches with the same size. In DeiT-S, there is only one such linear projection. These two backbones have the same number of parameters in their Transformer encoders.

Moreover, it can be seen from the experimental results that ViT-SAPS ($T_{split} = 3$) achieves much better segmentation performance for all the mechanical parts except bearing sleeve (Type 1) (C5) than ViT-SAPS ($T_{split} = 15$). This is because when T_{split} is smaller, the segmentation grain is finer, and thus the ViT-SAPS model can capture more detail information in the images.

2) ANALYSIS OF THE PROPOSED MODEL'S LOCALITY

For Dataset 1, the difference between the training set and the test set is only the shooting angles of the images. The models can learn all the assembly stages of all the products

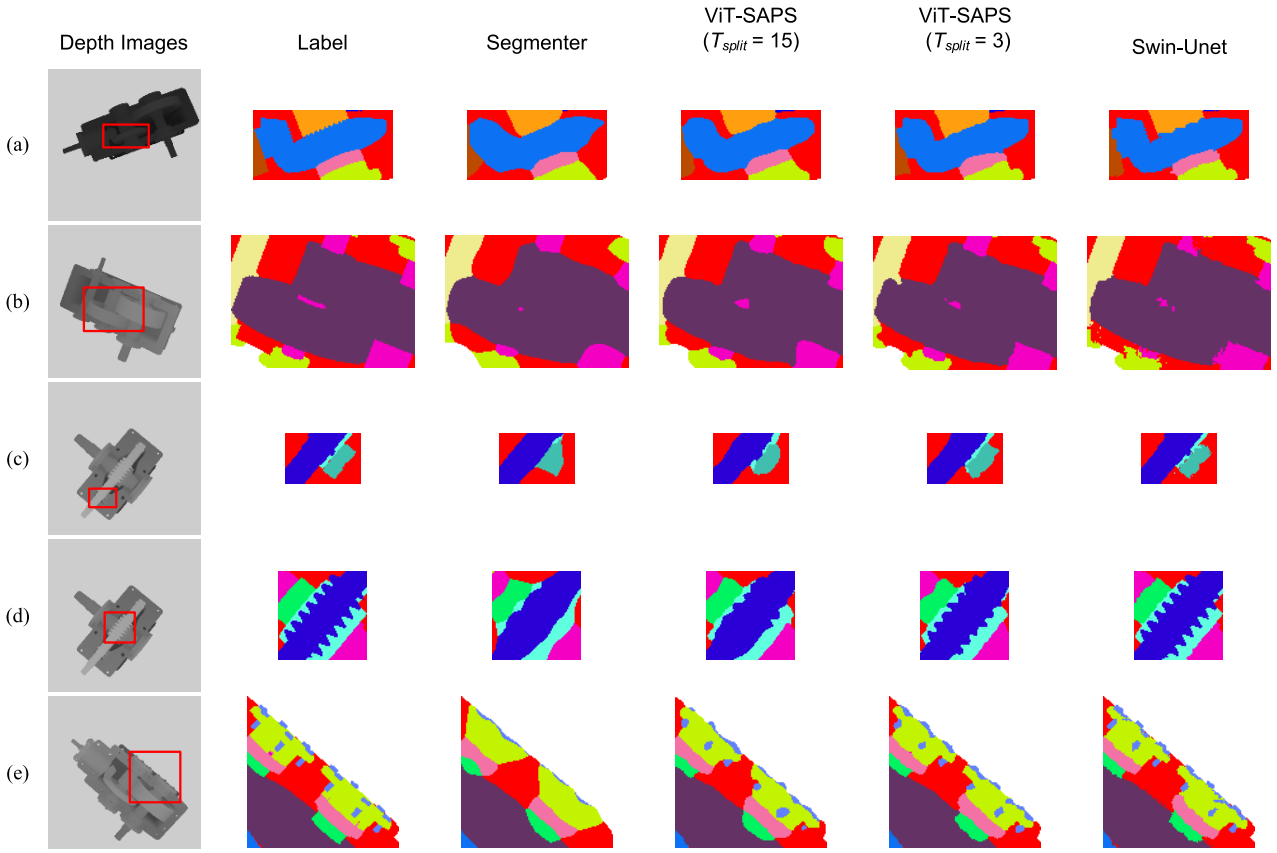


FIGURE 8. Segmentation performance of the pure Transformer models on the partial details of four mechanical products.

from the training set, and thus they are very familiar with the assembly positions of all the parts in each assembly stage. Therefore, the models seldom misjudge part types. The most important factor that affects a model's performance on this dataset is its ability to segment part edges, i.e., its locality. Among all the models tested in the experiments, the top five in terms of mIoU are GCAU-Net (hybrid model of CNN and attention mechanism), PSPNet (CNN-based), our ViT-SAPS ($T_{split} = 3$) (pure Transformer), U-Net (CNN-based), and Swin-Unet (pure Transformer). These experimental results indicate that the locality of ViT-SAPS is slightly worse than that of some CNN-based models and hybrid models, whereas it is at a relatively advanced level among the pure Transformer models.

Fig. 8 shows the segmentation performance of the pure Transformer models for the partial details of the four mechanical products on the test set of Dataset 1. Specifically, Fig. 8(a) and 8(b) show the details of gear connections and the cases where gears are shielded by each other, while Fig. 8(c), 8(d), and8(e) show the details of some tiny parts, namely oil scraper (C15), bearing sleeve (Type 2) (C6), bearing end-shield (C3), and bolt (C8). It can be seen from these figures that ViT-SAPS has much stronger detail segmentation ability than Segmenter. The 16×16 patch size makes Segmenter almost impossible to recognize extremely tiny parts like bolts. However, ViT-SAPS overcomes this difficulty by splitting the image regions with these tiny parts into much smaller patches. Moreover, it can be seen from these figures that reducing T_{split} can further enhance the detail segmentation ability of ViT-SAPS. Compared with Swin-Unet, our ViT-SAPS achieves smoother segmentation of part edges. When using Swin-Unet, burrs and even debris often appear on part edges. This phenomenon rarely occurs in practice. In contrast, the segmentation maps generated by ViT-SAPS are much closer to reality. Overall, it can be concluded that the self-adaptive patch splitting mechanism is a powerful method to enhance the locality of vision Transformers.

3) ANALYSIS OF THE PROPOSED MODEL'S GLOBALITY

The test set of Dataset 2 contains assembly stages that the model cannot learn from the training set. The model judges the type of a part by synthesizing the part's characteristics and its surrounding environment. Therefore, the globality of the model is crucial. The experimental results in Table 4 demonstrate that our ViT-SAPS ($T_{split} = 3$) achieves the best mIoU of 62.64%. In Dataset 2, there are four types of parts that are included in the training set but are not included in the test set, namely sleeve (C7), cylindrical helical gear shaft (C13), oil scraper (C15), and worm (C16). All the models in the experiment mistakenly believe that the test set contains sleeve (C7) and oil scraper (C15). Segmenter and ViT-SAPS can accurately determine that the test set does not contain

cylindrical helical gear shaft (C13) and worm (C16), whereas other models cannot. Similar to Segmenter and ViT-SAPS, Swin-Unet is also a pure Transformer model. However, it may lose some globality because it adopts a hierarchical structure similar to CNN, resulting in the misjudgment of cylindrical helical gear shaft (C13) and worm (C16). In the practical production process, such misjudgment has a fatal impact on the quality of mechanical assembly monitoring. Moreover, it can be seen from the experimental results that for bolt (C8), cylindrical spur gear (C9), and bevel gear (C11), the segmentation IoU of the models with convolution on Dataset 2 is much lower than that on Dataset 1, whereas the pure Transformer models do not have this problem. To sum up, in mechanical assembly semantic segmentation tasks, pure Transformer models are more stable and reliable than models with convolution due to their excellent globality. ViT-SAPS overcomes the poor locality of existing vision Transformers without losing globality, thus achieving a pretty good locality-globality trade-off.

V. CONCLUSION

This study proposes a novel vision Transformer called ViT-SAPS which is able to perceive the detail information in the image. ViT-SAPS can split the detailed and non-detailed regions of the image into patches with different sizes and pay finer-grained attention on the image regions where the detail information locates. Information distribution in different regions of a mechanical assembly image is extremely uneven. Hence, ViT-SAPS is suitable for the mechanical assembly semantic segmentation task and can provide solutions to automated mechanical assembly monitoring. The experimental results indicate that the adaptive patch splitting mechanism of ViT-SAPS is superior to the common fixed-size patch splitting mechanism in processing images in which the information quantity distribution is uneven, thus meeting the requirements of mechanical assembly semantic segmentation. The experimental results also show that ViT-SAPS performs well in both locality and globality, and therefore it can be extended to solve other image segmentation problems.

One limitation of our method is that the patch distribution in the input images is different from each other, so ViT-SAPS needs to consider each image independently when performing patch splitting and bilinear interpolation. Since this complex work is not suitable for GPU parallel processing, this paper uses CPU to complete this work, which is time-consuming and becomes the bottleneck of the model. The time cost of segmenting a single image is on the order of 1 second. This time cost can meet the needs of semantic segmentation in assembly monitoring, but it cannot meet the needs of other applications with high real-time requirements. Therefore, the acceleration of model processing is the key issue of future work. In future work, we will promote ViT-SAPS to other datasets and computer vision tasks, such as image classification and object detection, and then evaluate its performance. Meanwhile, we will popularize the idea of self-adaptive patch size to other existing vision Transformer models. For example, hierarchical models like PVT and Swin Transformer.

REFERENCES

[1] M. Kim, W. Choi, B.-C. Kim, H. Kim, J. H. Seol, J. Woo, and K. H. Ko, "A vision-based system for monitoring block assembly in shipbuilding," *Comput.-Aided Des.*, vol. 59, pp. 98–108, Feb. 2015, doi: 10.1016/j.cad.2014.09.001.

[2] C. Chen, C. Zhang, T. Wang, D. Li, Y. Guo, Z. Zhao, and J. Hong, "Monitoring of assembly process using deep learning technology," *Sensors*, vol. 20, no. 15, p. 4208, Jul. 2020, doi: 10.3390/s20154208.

[3] C. Chen, T. Wang, D. Li, and J. Hong, "Repetitive assembly action recognition based on object detection and pose estimation," *J. Manuf. Syst.*, vol. 55, pp. 325–333, Apr. 2020, doi: 10.1016/j.jmsy.2020.04.018.

[4] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, Jul. 2022, doi: 10.1016/j.neucom.2022.01.005.

[5] R. Azad, M. T. Al-Antary, M. Heidari, and D. Merhof, "TransNorm: Transformer provides a strong spatial normalization mechanism for a deep segmentation model," *IEEE Access*, vol. 10, pp. 108205–108215, 2022, doi: 10.1109/ACCESS.2022.3211501.

[6] Y. Wu, K. Liao, J. Chen, J. Wang, D. Z. Chen, H. Gao, and J. Wu, "D-former: A U-shaped dilated transformer for 3D medical image segmentation," *Neural Comput. Appl.*, vol. 35, no. 2, pp. 1931–1944, Jan. 2023, doi: 10.1007/s00521-022-07859-1.

[7] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," 2021, *arXiv:2105.05537*.

[8] Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan, and Z. He, "A survey of visual transformers," 2021, *arXiv:2111.06091*.

[9] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6881–6890.

[10] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7262–7272.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[13] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1529–1537.

[14] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1377–1385.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Munich, Germany, 2015, pp. 234–241.

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*.

[18] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.

[19] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2881–2890.

[20] L. C. Chen, G. Papandreou, and I. Kokkinos, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Jun. 2018, doi: 10.1109/TPAMI.2017.2699184.

[21] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, "PSANet: Point-wise spatial attention network for scene parsing," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, 2018, pp. 267–283.

[22] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3146–3154.

[23] H. Liu, Y. Fu, S. Zhang, J. Liu, Y. Wang, G. Wang, and J. Fang, "GCHA-net: Global context and hybrid attention network for automatic liver segmentation," *Comput. Biol. Med.*, vol. 152, Jan. 2023, Art. no. 106352, doi: 10.1016/j.compbiomed.2022.106352.

[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K., 2020, pp. 213–229.

[25] S. Li, X. Chen, D. He, and C.-J. Hsieh, "Can vision transformers perform convolution?" 2021, *arXiv:2111.01353*.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[27] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[28] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.

[29] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 568–578.

[30] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," 2021, *arXiv:2107.00641*.

[31] L. Yuan, Q. Hou, Z. Jiang, J. Feng, and S. Yan, "VOLO: Vision outlooker for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 6575–6586, May 2023, doi: 10.1109/TPAMI.2022.3206108.

[32] P. Sheng, Y. Shi, X. Liu, and H. Jin, "LSNet: Real-time attention semantic segmentation network with linear complexity," *Neurocomputing*, vol. 509, pp. 94–101, Oct. 2022, doi: 10.1016/j.neucom.2022.08.049.

[33] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. H. Tay, J. Feng, and S. Yan, "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 558–567.

[34] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 15908–15919.

[35] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[36] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin transformer V2: Scaling up capacity and resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 12009–12019.

[37] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 12077–12090.

[38] Y. Wang, R. Huang, S. Song, Z. Huang, and G. Huang, "Not all images are worth 16×16 words: Dynamic transformers for efficient image recognition," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 11960–11973.

[39] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 9355–9366.

[40] P. Zhang, X. Dai, J. Yang, B. Xiao, L. Yuan, L. Zhang, and J. Gao, "Multi-scale vision longformer: A new vision transformer for high-resolution image encoding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2998–3008.

[41] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your ViT? Data, augmentation, and regularization in vision transformers," 2021, *arXiv:2106.10270*.

[42] M. A. Oskoei and H. Hu, "A survey on edge detection methods," School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-506, Feb. 2010.

[43] A. Asjad and D. Mohamed, "A new approach for salt dome detection using a 3D multidirectional edge detector," *Appl. Geophys.*, vol. 12, no. 3, pp. 334–342, Oct. 2015, doi: 10.1007/s11770-015-0512-2.

HAITAO DONG received the B.Eng. degree in communication engineering from the University of Electronic Science and Technology of China, in 2011, and the M.S. and Ph.D. degrees in signal and information processing from the Institute of Acoustics, Chinese Academy of Sciences, in 2016. He is currently a Lecturer with the School of Information and Control Engineering, Qingdao University of Technology. His research interests include computer vision and deep learning.

CHENGJUN CHEN received the B.Eng. and Ph.D. degrees from the School of Mechanical Engineering, Shandong University, China, in 2003 and 2008, respectively. He is currently a Professor with the School of Mechanical and Automotive Engineering, Qingdao University of Technology, China. His major research interests include virtual reality and augmented reality in industrial applications.

JINLEI WANG received the B.Eng. and M.S. degrees in mechanical engineering from the Qilu University of Technology. He is currently pursuing the Ph.D. degree with the School of Mechanical and Automotive Engineering, Qingdao University of Technology. His research interests include image processing, computer vision, and deep learning.

FEIXIANG SHEN received the bachelor's degree from the School of Mechanical and Automotive Engineering, Qingdao University of Technology, in 2020, where he is currently pursuing the master's degree. His research interests include image processing and defect detect.

YONG PANG received the bachelor's degree in communication engineering and the master's degree in signal and information processing from Shandong University, in 2012 and 2015, respectively. She is currently pursuing the Ph.D. degree with the School of Mechanical and Automotive Engineering, Qingdao University of Technology. She is an Engineer with the School of Information and Control Engineering, Qingdao University of Technology.