Journal of Computational Design and Engineering, 2022, 9(3), 1123–1134

https://doi.org/10.1093/jcde/qwac049
Journal homepage: www.jcde.org
Advance Access Publication Date: 24 May 2022

RESEARCH ARTICLE

Real-time surgical tool detection in computer-aided surgery based on enhanced feature-fusion convolutional neural network

Kaidi Liu[1], Zijian Zhao[1,*], Pan Shi[1], Feng Li[2] and He Song[1]

[1]School of Control Science and Engineering, Shandong University, Jinan 250061, China and [2]Department of General Surgery, Qilu Hospital of Shandong University, Jinan 250012, China

*Corresponding author. E-mail: zhaozijian@sdu.edu.cn

Abstract

Surgical tool detection is a key technology in computer-assisted surgery, and can help surgeons to obtain more comprehensive visual information. Currently, a data shortage problem still exists in surgical tool detection. In addition, some surgical tool detection methods may not strike a good balance between detection accuracy and speed. Given the above problems, in this study a new Cholec80-tool6 dataset was manually annotated, which provided a better validation platform for surgical tool detection methods. We propose an enhanced feature-fusion network (EFFNet) for real-time surgical tool detection. FENet20 is the backbone of the network and performs feature extraction more effectively. EFFNet is the feature-fusion part and performs two rounds of feature fusion to enhance the utilization of low-level and high-level feature information. The latter part of the network contains the weight fusion and predictor responsible for the output of the prediction results. The performance of the proposed method was tested using the ATLAS Dione and Cholec80-tool6 datasets, yielding mean average precision values of 97.0% and 95.0% with 21.6 frames per second, respectively. Its speed met the real-time standard and its accuracy outperformed that of other detection methods.

Keywords: feature fusion; convolutional neural network; real-time surgical tool detection; computer-assisted surgery

List of symbols

a_i :	Intersection over union of prediction object bounding box and ground truth.
A^c :	Minimum rectangular area surrounding prediction object bounding box and ground truth.
b_i :	Predicted value.
$\hat{b}_i$:	Prediction confidence obtained via sigmoid function.
c_{ij} :	Whether there is an object of class j in the prediction object bounding box i.
d_{ij} :	Predicted value.
$\hat{d}_{ij}$:	Object probability obtained by sigmoid function.
L:	Total loss.
$L_{conf}(a, b)$:	Confidence loss.
$L_{cla}(c, d)$:	Classification loss.
$L_{loc}(GIoU)$:	Location loss.
N:	Number of positive and negative samples.
N_{pos} :	Number of positive samples.
u:	Union of ground truth and prediction object bounding box.
λ_1:	Equilibrium coefficient.
λ_2:	Equilibrium coefficient.
λ_3:	Equilibrium coefficient.

Received: 13 December 2021; **Revised:** 24 April 2022; **Accepted:** 14 May 2022

1. Introduction

In minimally invasive surgery (MIS), it is crucial to accurately locate key anatomical positions (Stoyanov, 2012). However, the narrow operating space and limited visual field increase the possibility of tissue damage, so more visual information on tissues, organs, and surgical tools is required. The emergence of computer-assisted surgery (CAS) provided a breakthrough solution to the bottleneck of MIS. Surgical tool detection is a key technology in CAS, and can help surgeons to obtain more visual information and improve surgical navigation. Real-time surgical tool detection enables real-time surgical video analysis (Jin *et al.*, 2018b, 2020; Yengera *et al.*, 2018), alerts clinicians to potential complications, optimizes surgical scheduling, and provides valid and objective feedback on the surgeons' surgical skills (Jin *et al.*, 2018a). Therefore, we focus on detecting surgical tools more accurately under real-time standard.

Many surgical tool detection methods have been proposed and are roughly divided into three categories: hardware, image-processing, and deep-learning methods. Initially, such hardware-based methods as optical information (Joskowicz *et al.*, 1998) and magnetic field (Fried *et al.*, 1997; Yang *et al.*, 2009) were used to detect surgical tools. Although hardware-based methods are simple and stable, they require modification of instruments and expensive tracking equipment. Later, image-processing methods were widely used. These methods located surgical tools based on visual cues from the tip of surgical tools such as colour, gradient, texture, and shape (Lee *et al.*, 1994; Krupa *et al.*, 2003; Alsheakhali *et al.*, 2015). However, these methods take a long time to process, and their application scope is limited (Zhao *et al.*, 2017, 2019a).

With the continuous development of artificial intelligence, surgical tool detection methods based on deep learning have become the mainstream. Faster region-based convolutional neural network (R-CNN; Ren *et al.*, 2017) was adjusted to achieve the positioning of surgical tools and evaluate surgeons' operating techniques (Sarikaya *et al.*, 2017; Jin *et al.*, 2018a). You only look once (YOLO; Redmon *et al.*, 2016) was improved to realize surgical tool detection (Choi *et al.*, 2017). U-net (Ronneberger *et al.*, 2015) was improved to estimate the posture of surgical tools (Kurmann *et al.*, 2017; Laina *et al.*, 2017; Du *et al.*, 2018; Gao *et al.*, 2019; Ni *et al.*, 2019). Two CNNs were cascaded to achieve surgical tool presence detection and real-time surgical tool tracking (Wang *et al.*, 2017, 2019b; Zhao *et al.*, 2019a, 2019b). Twinanda *et al.* (2017) combined a support vector machine and the hidden Markov model, and fine-tuned AlexNet (Krizhevsky *et al.*, 2012) to simultaneously perform surgical tool presence detection and phase recognition. Mishra *et al.* (2017) and Al Hajj *et al.* (2018) introduced a recurrent neural network to detect tools in surgery videos. Vania *et al.* (2019) combined a CNN and fully convolutional network (Shelhamer *et al.*, 2017), and used class redundancy as a soft constraint to realize automatic spine segmentation from computer-tomography images. Liu *et al.* (2020) combined an anchor-free CNN (Zhou *et al.*, 2019) and a stacked hourglass network (Newell *et al.*, 2016) to achieve real-time surgical tool detection. Shi *et al.* (2020) introduced an attention mechanism (Hu *et al.*, 2020) to realize real-time surgical tool detection. Vania and Lee (2021) proposed a multistage optimization mask R-CNN (He *et al.*, 2017) to realize automatic intervertebral disc instance segmentation. Lee *et al.* (2021) used YOLOv3 (Redmon & Farhadi, 2018) to achieve detect surroundings of a ship. Nwoye *et al.* (2019) and Vardazaryan *et al.* (2018) adopted weak supervision to weaken the influence of insufficient training samples and increase the generalization ability of the model.

Combined with a classical neural network, these methods realize many medical image tasks, including surgical tool detection, surgical tool presence detection, posture estimation, tracking, localization, instance segmentation, and surgical evaluation. This study focuses on surgical tool detection. The EndoVis Challenge dataset (Du *et al.*, 2018) and ATLAS Dione dataset (Sarikaya *et al.*, 2017) are public datasets used for surgical tool detection. However, a data shortage problem still exists in surgical tool detection. In addition, many methods cannot meet the real-time standard or achieve relatively good accuracy under real-time conditions.

In this work, we studied the problems described earlier. The main contributions are summarized as follows.

(i) We manually annotated a new dataset, the Cholec80-tool6 dataset, which originated from the Cholec80 dataset (Twinanda *et al.*, 2017). The video sequence of this dataset presents a more realistic surgical environment and has more practical significance for the validation of relevant methods.
(ii) By optimizing EfficientNetV2 (Tan & Le, 2021), we proposed a simpler and faster backbone called FENet20, which extract feature more effectively and improved the detection accuracy.
(iii) We designed an enhanced feature-fusion network (EFFNet) to obtain better detection results. EFFNet performed two rounds of feature fusion to enhance the utilization of low- and high-level feature information.

We verified the performance of the proposed method via the ATLAS Dione and Cholec80-tool6 datasets, and the method met the real-time standard and had a higher detection accuracy than 10 other detection methods.

2. Methodology

2.1. Dataset

The ATLAS Dione dataset (Sarikaya *et al.*, 2017) contains 99 video clips of 10 surgeons from the Roswell Park Cancer Institute (Buffalo, USA) performing six different surgical tasks on the da Vinci Surgical System. This dataset was a phantom setting and had a large number of samples (Liu *et al.*, 2020; Shi *et al.*, 2020). Each frame was annotated with the spatial coordinates of surgical tools, and the resolution was 854 × 480. We used 90 video clips (20 491 frames) for the training and nine video clips (1976 frames) for the testing in this study.

Real-life MIS contains many disturbance factors, such as motion blurring, high deformation, lens fogging, tools with occlusion, and missing parts. All of these disturbance factors pose a challenge to the surgical tool detection method. The ATLAS Dione dataset is a phantom setting and may lack some disturbance factors, e.g. lens fogging, and there is a gap between it and an actual surgical environment. Although this dataset has fewer types of tools and lacks some disturbance factors, it has a large number of samples, so we still used the ATLAS Dione dataset.

To enrich the training samples, under the supervision of surgeons, we took the data annotation of the ATLAS Dione dataset as the standard and manually annotated a new Cholec80-tool6 dataset, the video sequence of which came from the Cholec80 dataset. The Cholec80 dataset is a collection of 80 cholecystectomy videos performed by 13 surgeons in Strasbourg, France.

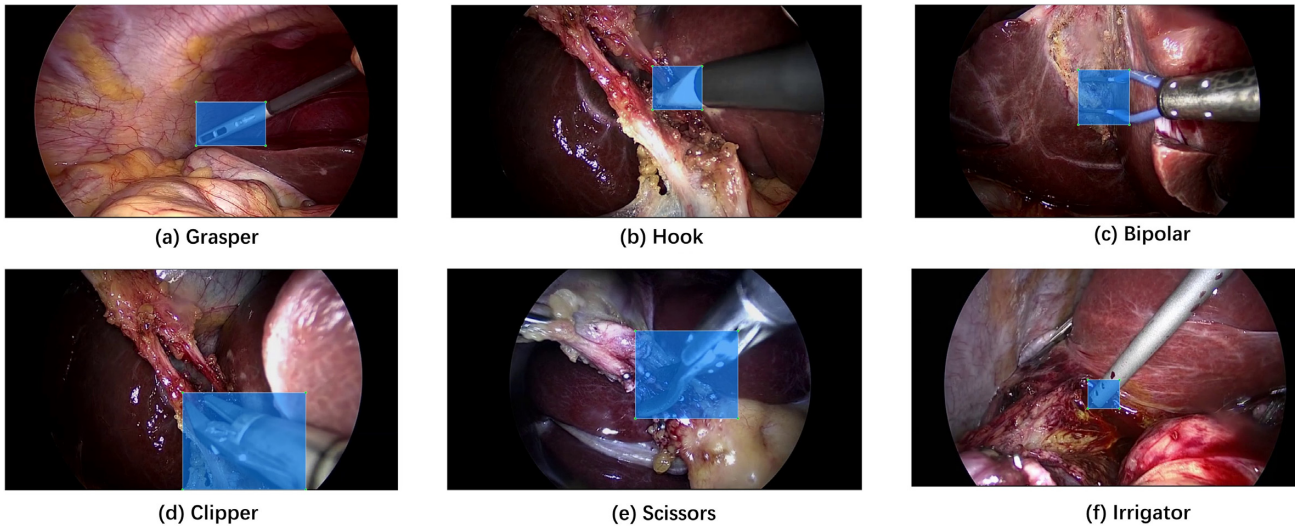

Figure 1: Examples of marking surgical tool tips.

Table 1: Number of annotated frames for each tool in m2cai16-tool, Cholec80-locations, and Cholec80-tool6 datasets.

	Number of annotated instances		
Tool	M2cai16-tool	Cholec80-locations	Cholec80-tool6
Grasper	923	2880	3226
Hook	308	1263	1514
Bipolar	350	579	663
Scissors	400	388	393
Clipper	400	400	477
Irrigator	485	485	400
Specimen bag	275	476	—
Total	3141	6471	6673
No. of frames	2532	4011	4013

Each frame was marked with the presence or absence of the tool, but not with its spatial coordinates.

We took 4013 frames from the Cholec80 dataset to mark the spatial coordinates of surgical tools. Since only the tip of the surgical tool could be seen in most minimally invasive surgical videos and the handles of different types of surgical tools were similar, we marked the tip of the surgical tool instead of the entire surgical tool itself, as shown in Fig. 1. The Cholec80-tool6 dataset consisted of six types of surgical tools (grasper, hook, bipolar, clipper, scissors, and irrigator), and the resolution of each frame was 854 × 480. We used 3270 frames for the training and 743 frames for the testing in this study.

The EndoVis Challenge (Du *et al.*, 2018), m2cai16-tool (Jin *et al.*, 2018a), Cholec80-locations (Shi *et al.*, 2020), and Cholec80-tool6 are all datasets for surgical tool detection. Unlike the ATLAS Dione dataset, they are taken from actual surgical environments. Data characteristics of the last three datasets are shown in Table 1. The EndoVis Challenge dataset has a smaller number of frames (1083 frames) and fewer types of surgical tools, and is less effective for the validation of surgical tool detection methods (Liu *et al.*, 2020; Shi *et al.*, 2020; Yang *et al.*, 2021). The Cholec80-tool6 dataset has more frames and annotated instances than the m2cai16-tool and Cholec80-locations datasets. This indicates that the Cholec80-tool6 dataset is richer in content.

If the m2cai16-tool, Cholec80-locations, and Cholec80-tool6 datasets are merged, then a dataset with a larger total number of frames will be produced. This large dataset is more conducive to the validation of the relevant methods. If one wants to use this large dataset, we suppose that it may be simpler to make changes to the way the methods read the data. It is probably time and labour consuming to modify the contents of the datasets for merging.

2.2. Data pre-processing

In surgical tool detection, the number of samples often fails to meet the training requirements, so it is necessary to use data augmentation. Common data augmentation methods include random horizontal flipping, random hue/saturation/value adjustment, random rotation, scaling, and translation. We used these common methods and added Mosaic data augmentation (Bochkovskiy *et al.*, 2020) in the training. Mosaic data augmentation reads four pictures each time, and then flips, zooms, and changes the colour gamut on the four pictures, and places them in four directions (upper left, upper right, lower left, and lower right). This process greatly enriches the dataset; in particular, random scaling adds many small objects.

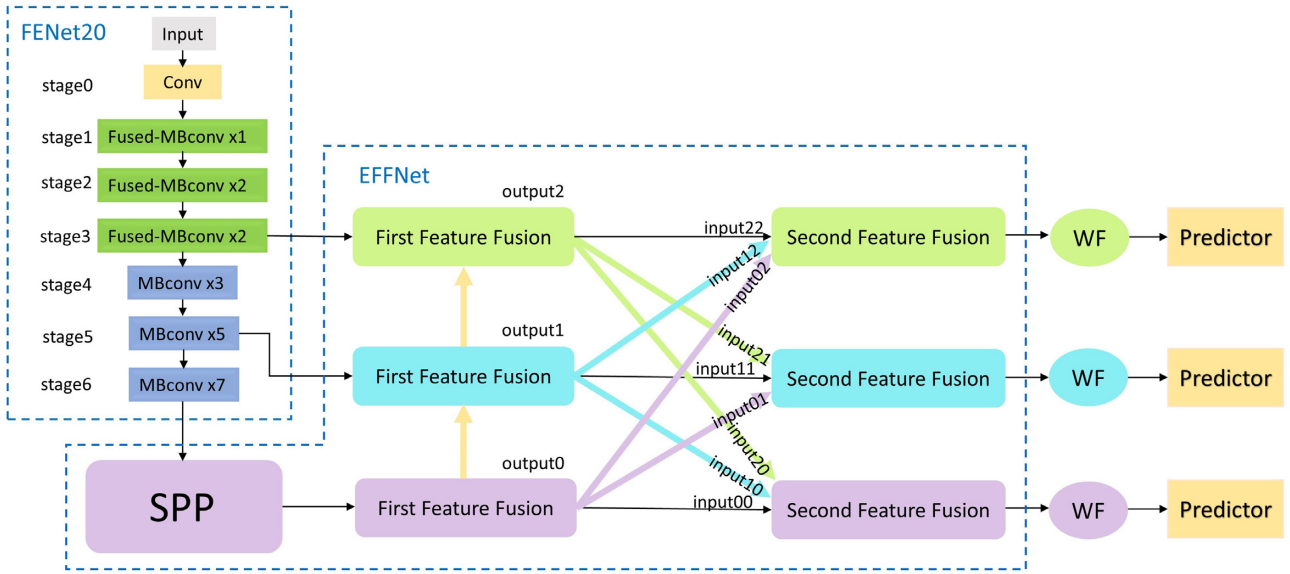

Figure 2: Overall network framework includes FENet20, EFFNet, WF, and predictor.

Table 2: Architecture of FENet20, in which *t* represents the expansion ratio (see Fig. 3 for an explanation), *s* the stride, *c* the output channels, *n* the number of times the operator is repeatedly stacked in the current stage, and output the output size of each stage when the input size is 384 × 384.

Stage	Operator	*t*	*s*	*c*	*n*	Output
0	Conv3 × 3	—	2	24	1	192 × 192
1	Fused-MBconv	1	1	24	1	192 × 192
2	Fused-MBconv	4	2	48	2	96 × 96
3	Fused-MBconv	4	2	64	2	48 × 48
4	MBconv	4	2	128	3	24 × 24
5	MBconv	6	1	160	5	24 × 24
6	MBconv	6	2	256	7	12 × 12

2.3. Network architecture

As shown in Fig. 2, the network architecture of the proposed method includes FENet20, EFFNet, weight fusion (WF), and a predictor. FENet20 is the backbone of the network and performs feature extraction. EFFNet is the feature-fusion part and performs two rounds of feature fusion. The latter part of the network contains the WF part and the predictor.

FENet20. Inspired by EfficientNetV2 (Tan & Le, 2021), we proposed a simpler and faster backbone called FENet20. Table 2 shows the architecture of FENet20, stage 0 was a common convolution, stages 1–3 contained the Fused-MBconv module, and stages 4–6 contained the MBconv module. The Fused-MBconv module contained 3 × 3 convolution, 1 × 1 convolution, and a dropout layer (Huang *et al.*, 2016). The MBconv module contained 1 × 1 convolution, 3 × 3 deep separable convolution, a squeeze-and-excitation (SE) module (Hu *et al.*, 2020), and a dropout layer. Figure 3 shows Fused-MBconv and MBconv modules.

EFFNet. We designed an EFFNet to obtain better detection results, as shown in Fig. 2. EFFNet includes two rounds of feature fusion.

First feature fusion: The spatial pyramid pooling (SPP) module used the max-pooling of four different scales to process the feature layer. The pooling core sizes were 13 × 13, 9 × 9, 5 × 5, and 1 × 1. This operation was intended to increase the receptive field and separate the most significant context features. In Fig. 2, the purple first feature-fusion module contained only three convolution layers. The blue and green first feature-fusion modules contained five convolution layers and one concatenation layer. The outputs of the three first feature-fusion modules are denoted as output 0, output 1, and output 2. Bold yellow arrows represent convolution and up-sampling operations. The first feature-fusion module is shown in Fig. 4.

Second feature fusion: In Fig. 2, the purple second feature-fusion module has three inputs, denoted as input 00, input 10, and input 20. Output 0 remained unchanged to obtain input 00, output 1 performed convolution to obtain input 10, while output 2 performed max-pooling and convolution to obtain input 20. In this way, the feature-map sizes of input 00, input 10, and input 20 were the same, as was the number of channels. The blue second feature-fusion module has three inputs, denoted as input 01, input 11, and input 21. Output 0 performed convolution and interpolation to obtain input 01, output 1 remained unchanged to obtain input 11, while output 2 performed convolution to obtain input 21. In this way, the feature-map sizes of input 01, input 11, and input 21 were the same, as was the number of channels. The green second feature-fusion module has three inputs, denoted as input 02, input 12, and input 22. Output 0 performed convolution and interpolation to obtain input 02, output 1 performed convolution and interpolation to obtain input 12, while output 2 remained unchanged to obtain input 22. In this way, the feature-map sizes of input 02, input 12, and input 22 were the same, as

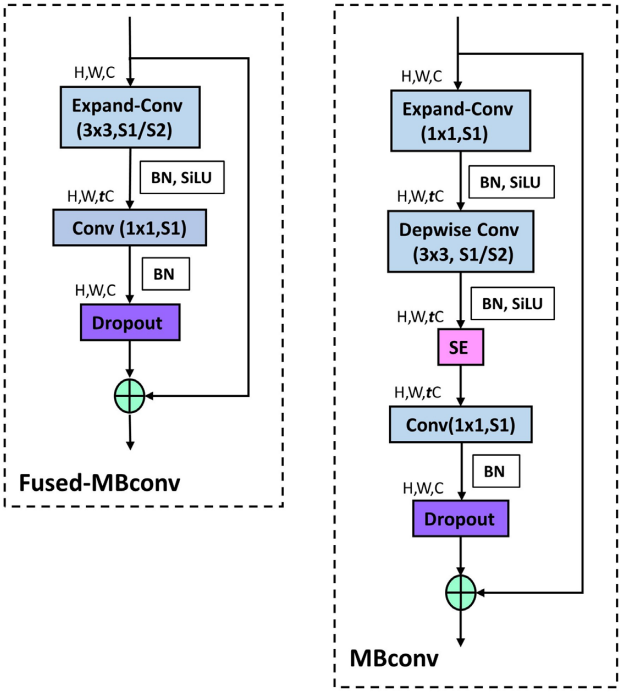

Figure 3: Fused-MBconv (left-hand panel) and the MBconv (right-hand panel) modules. Expand-Conv is able to expand the channel of the input feature layer, and t represents the expansion ratio. SiLU (sigmoid linear unit) represents the activation function (Hendrycks & Gimpel, 2016) and s the stride.

was the number of channels. The second feature-fusion module is also shown in Fig. 4, where input 0∗ represents input 00/01/02, input 1∗ represents input 10/11/12, and input 2∗ represents input 20/21/22.

WF and predictor. Inspired by Liu *et al.* (2019) and Wang *et al.* (2019a), we added the WF module after the second feature-fusion module. The WF module performed a softmax operation on the input feature layer and took out the three weights. These three weights correspond to input 0∗, input 1∗, and input 2∗. Then, the WF module performed a weighting computation to obtain the feature layer after the fusion of weights. The predictor used the YOLO Head from YOLOv3 (Redmon & Farhadi, 2018). The predictor contained a 3×3 convolution layer and a 1×1 convolution layer. The 3×3 convolution layer doubled the number of channels of the input feature layer, while the 1×1 convolution layer reduced the number of channels to $3 \times (\text{num\_class} + 4 + 1)$, where num_class represents the number of classes of predicted objects.

2.4. Loss function

The design of loss referred to YOLOv1 (Redmon *et al.*, 2016), YOLOv2 (Redmon & Farhadi, 2017), and YOLOv3 (Redmon & Farhadi, 2018). The loss of network altogether included confidence, classification, and location losses, and the total loss is expressed as follows:

$$L = \lambda_1 L_{\text{conf}}(a, b) + \lambda_2 L_{\text{cla}}(c, d) + \lambda_3 L_{\text{loc}}(\text{GIoU}), \tag{1}$$

where λ_1, λ_2, and λ_3 are the equilibrium coefficients, while $L_{\text{conf}}(a, b)$, $L_{\text{cla}}(c, d)$, and $L_{\text{loc}}(\text{GIoU})$ represent confidence, classification, and location losses, respectively.

Our model has three prediction feature layers. Each prediction feature layer predicts bounding boxes at three different scales. Confidence represents the probability that there is an object in the predicted bounding box. Confidence loss is expressed using a binary cross-entropy function:

$$L_{\text{conf}}(a, b) = -\frac{\sum_i (a_i \ln(\hat{b}_i) + (1 - a_i)\ln(1 - \hat{b}_i))}{N}, \tag{2}$$

where $a_i \in [0, 1]$ is the intersection over union (IoU) of the prediction object bounding box and ground truth, b_i is the predicted value, $\hat{b}_i = Sigmoid(b_i)$ is the prediction confidence obtained by the sigmoid function, and N is the number of positive and negative samples.

Each bounding box generates C conditional class probabilities, with C as the number of classes of predicted objects. Each class is subjected to independent logistic regression to obtain an object probability in the range 0–1. In training, each class uses a binary cross-entropy function:

$$L_{\text{cla}}(c, d) = -\frac{\sum_{i \in \text{pos}} \sum_{j \in \text{cla}} (c_{ij} \ln(\hat{d}_{ij}) + (1 - c_{ij})\ln(1 - \hat{d}_{ij}))}{N_{\text{pos}}}, \tag{3}$$

where $c_{ij} \in \{0, 1\}$ represents whether there is an object of class j in the prediction object bounding box i, d_{ij} is the predicted value, $\hat{d}_{ij} = \text{Sigmoid}(d_{ij})$ represents the object probability obtained by the sigmoid function, and N_{pos} is the number of positive samples.

For the design of the location loss, we drew inspiration from YOLOv3. The difference was that we used generalized intersection over union (GIoU) loss (Rezatofighi *et al.*, 2019) instead of mean-square-error loss:

$$\text{GIoU} = IoU - \frac{A^c - u}{A^c}, \tag{4}$$

$$L_{\text{GIoU}} = 1 - \text{GIoU}, \tag{5}$$

where A^c is the minimum rectangular area surrounding the ground truth and prediction object bounding box, while u represents the union of the ground truth and prediction object bounding box. If $-1 \le \text{GIoU} \le 1$, $L_{\text{GIoU}} = 1 - \text{GIoU}$, yielding $0 \le L_{\text{GIoU}} \le 2$. The GIoU considers that the loss of IoU is the same when there is no overlap between the prediction object bounding box and the ground truth. Therefore, GIoU introduces the concept of A^c, but when the prediction object bounding box and the ground truth contain each other, GIoU in Equation (4) degenerates to IoU.

3. Experiment

3.1. Experimental set-up and implementation

The experimental platform was the Ubuntu 18.04 LTS operating system. The experimental environment included Python 3.8, CUDA 10.0, and PyTorch 1.7.0. The accelerator was an NVIDIA GeForce GTX TITAN X graphical processing unit. The specific implementation of the experiment was as follows. In training, we used a multiscale training method. The basic size of 384×384 was adjusted every 10 batches within the range 67–150%. Then, data-augmentation methods described in Section 2.2 were applied to enrich the training samples. In testing, the sample size was 384×384. It is worth noting that Mosaic data augmentation was switched on in training and off in testing. The optimizer parameter configuration implied an initial learning rate of 1×10^{-3}, momentum of 0.937, weight decay of 5×10^{-4}, batch size of 8, and training epoch of 200.

3.2. Detection results and comparative tests

The proposed method was validated using the ATLAS Dione and Cholec80-tool6 datasets. The ATLAS Dione dataset includes two

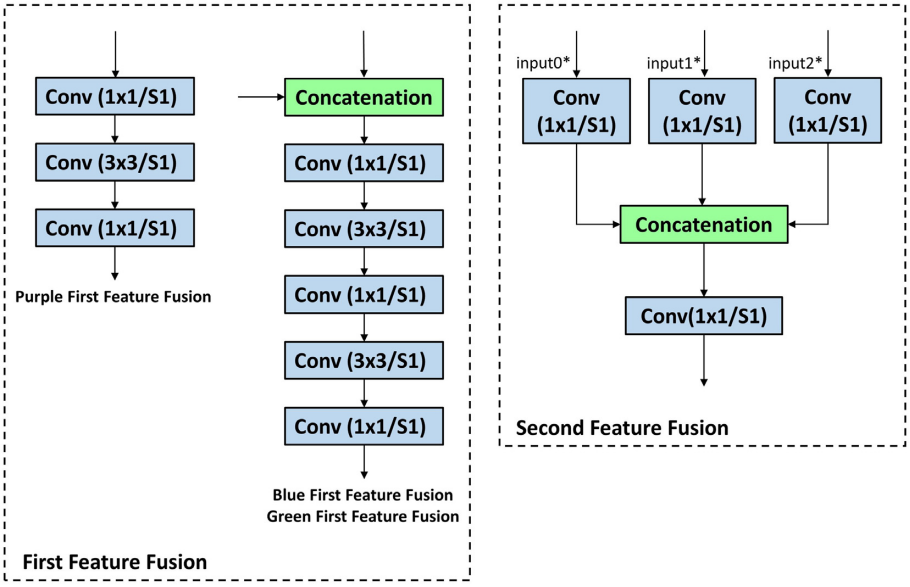

Figure 4: First (left-hand panel) and second (right-hand panel) feature-fusion modules, ∗ represents 0/1/2.

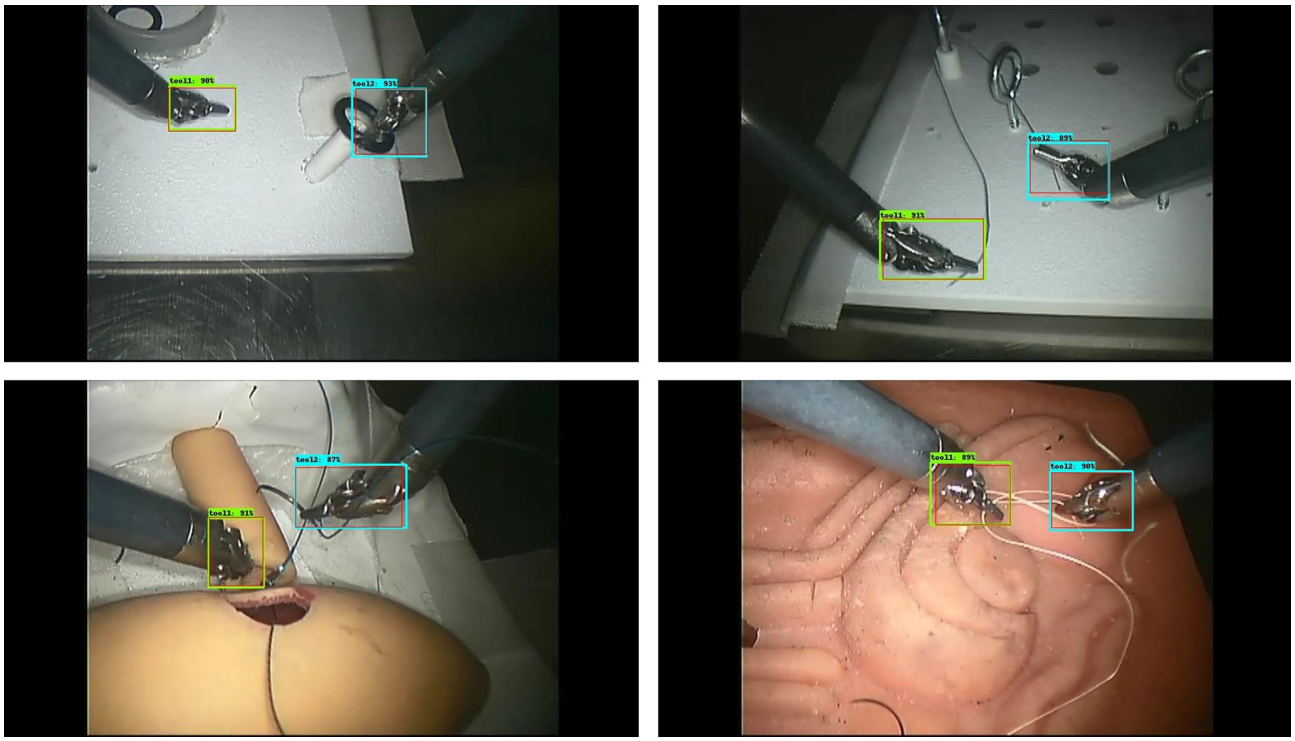

Figure 5: Detection results for ATLAS Dione dataset; red boxes represent ground truth.

types of surgical tools (tool 1 and tool 2). The Cholec80-tool6 dataset includes six types of surgical tools (grasper, hook, bipolar, clipper, scissors, and irrigator). As shown in Figs. 5 and 6, good detection results were obtained for tool 1, tool 2, and six types of surgical tools. As shown in Table 1, the number of annotated instances is high for grasper, hook, and bipolar, and low for scissors, clipper, and irrigator. Therefore, the detection results of grasper, hook, and bipolar were better than those of scissors, clipper, and irrigator in Fig. 6.

To assess its feasibility, the proposed method was compared with seven classical object detection methods and three surgical tool detection methods. The former comprised one two-stage method (Faster R-CNN; Ren *et al.*, 2017), five one-stage methods [single shot multibox detector (SSD; Liu *et al.*, 2016), RetinaNet (Lin *et al.*, 2020), YOLOv3 (Redmon & Farhadi, 2018), RefineDet (Zhang *et al.*, 2018), and YOLOv4 (Bochkovskiy *et al.*, 2020)], and one anchor-free method [fully convolutional one-stage (FCOS; Tian *et al.*, 2019)]. The three surgical tool detection methods comprised those of Liu *et al.*

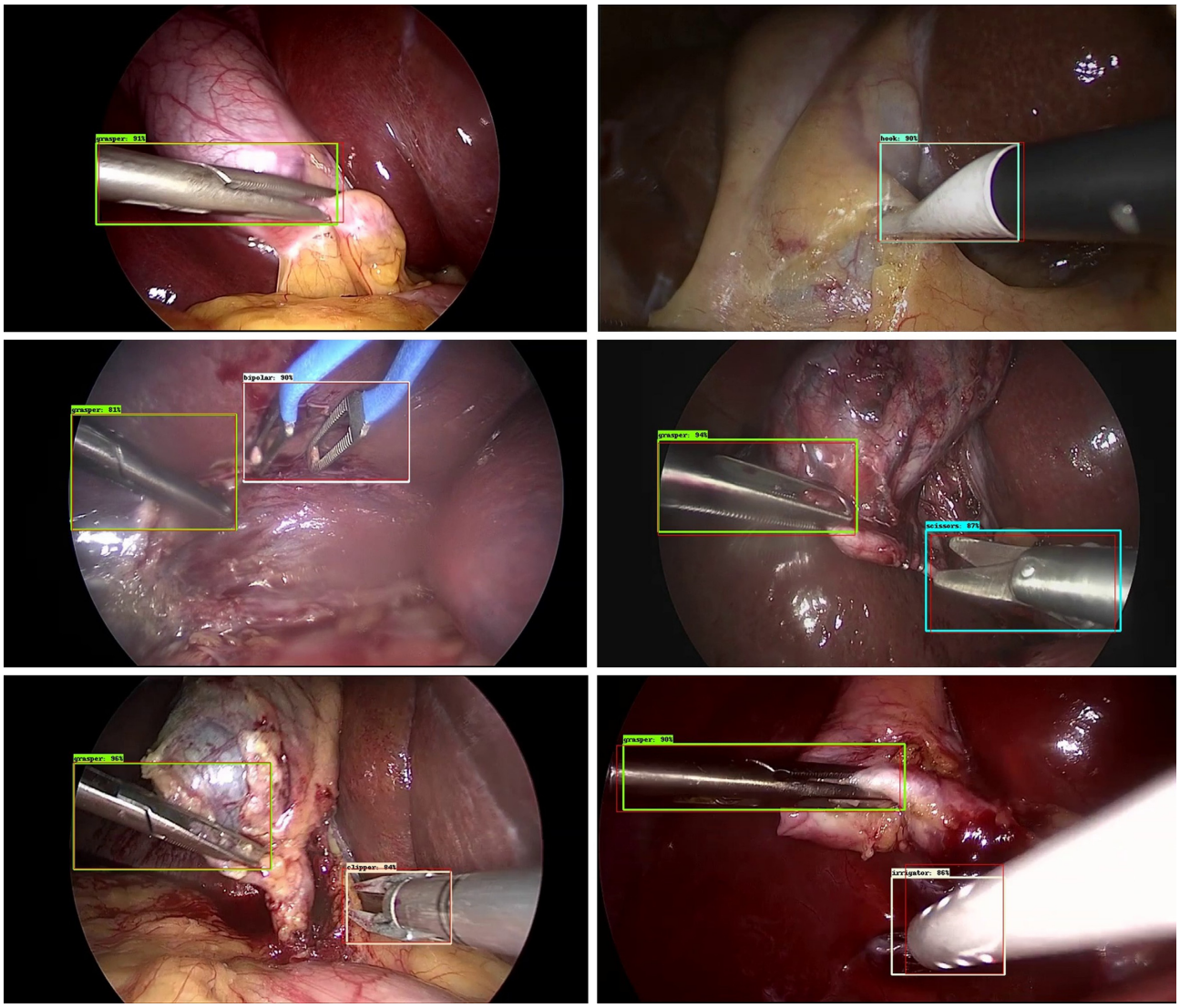

Figure 6: Detection results for Cholec80-tool6 dataset; red boxes represent ground truth.

(2020), Shi *et al.* (2020), and Yang *et al.* (2021). These methods were applied to the ATLAS Dione and Cholec80-tool6 datasets, the results of which are summarized in Tables 3 and 4, respectively.

Figure 7 intuitively shows the mAP for all methods on the ATLAS Dione dataset. Since the ATLAS Dione dataset is a phantom setting, the mAP of some methods is relatively close. The proposed method achieved the highest accuracy compared with the other methods. Figure 8 intuitively shows the FPS for all methods on the ATLAS Dione dataset. If the FPS value is ≥20, then the surgical tool detection method meets the real-time standard (Liu *et al.*, 2020). The proposed method runs in real time at 21.6 FPS. Although the detection speed of the proposed method is lower than that of several of the other methods, it still meets the real-time standard.

Compared with the ATLAS Dione dataset, the Cholec80-tool6 dataset corresponded to an actual surgical environment, featuring six types of surgical tools and provided a more challenging test for detection methods. As seen in Table 4, compared with the other detection methods, the proposed method not only had the highest accuracy, but also met the real-time standard. From the accuracy obtained for each type of surgical tool, it can be seen that the accuracy of grasper, hook, and bipolar was higher than that of clipper, scissors, and irrigator, which complies with the results depicted in Fig. 6. Figures 7 and 8 intuitively show the mAP and FPS, respectively, for all methods on the Cholec80-tool6 dataset.

We analysed the results of the comparative tests. The detection methods in comparison tests can be divided into anchor-based and -free methods. The method of Liu *et al.* (2020) is an anchor-free CNN and the backbone is Lightweight Hourglass network. Therefore, their method is simple and fast. There is a small gap between the accuracy of the proposed method and that of Liu *et al.* (2020) on the ATLAS Dione dataset, and, in the Cholec80-tool6 dataset, a relatively obvious gap exists between the accuracy of the proposed method and that of Liu *et al.* (2020).

The above-mentioned gaps exist because the proposed method is anchor-based, and the anchor box used constrains the predicted-object range and adds a priori experience of size, so as to realize multiscale learning (Redmon & Farhadi, 2018). In addition, FENet20 and EFFNet make the proposed network architecture more complex and have a greater number of layers. There-

Table 3: Results for all methods applied to ATLAS Dione dataset. Columns 3 and 4 indicate mean average precision (mAP) for two types of surgical tools, where mAP corresponds to IoU = 0.5 and last column indicates frames per second (FPS).

Method	Backbone	Tool 1	Tool 2	mAP (%)	FPS
YOLOv3	DarkNet53	93.3	90.2	91.7	31.7
SSD	Vgg16	89.5	97.0	93.2	26.3
Faster R-CNN	Vgg16	89.7	96.9	93.3	14.4
RefineDet	Vgg16	89.2	97.8	93.5	33.3
Shi *et al.* (2020)	Vgg16	91.2	97.0	94.1	55.0
RetinaNet	Resnet50	95.4	94.8	95.1	13.4
Yang *et al.* (2021)	Ghost module + CSP module	93.0	97.0	95.1	36.6
YOLOv4	CSPDarknet53	95.4	95.8	95.6	22.7
FCOS	Resnet50	95.0	97.1	96.1	28.5
Liu *et al.* (2020)	Lightweight Hourglass	95.2	97.4	96.3	36.5
Proposed	FENet20	95.6	98.4	97.0	21.6

Table 4: Results for all methods applied to Cholec80-tool6 dataset. Columns 2–7 indicate mAP for six types of surgical tools, where mAP corresponds to IoU = 0.5 and last column indicates FPS.

Method	Grasper	Hook	Bipolar	Clipper	Scissors	Irrigator	mAP (%)	FPS
YOLOv3	96.7	97.7	93.4	91.6	84.7	67.8	88.6	31.7
Faster R-CNN	96.5	96.7	91.8	94.5	88.0	72.7	90.0	14.4
SSD	90.7	90.6	92.9	90.9	88.2	91.2	90.8	26.3
RefineDet	90.6	90.9	100.0	90.9	90.2	90.1	92.1	33.3
Shi *et al.* (2020)	90.2	99.8	91.2	91.0	90.9	89.5	92.1	55.0
RetinaNet	98.8	98.6	93.8	93.0	84.5	85.3	92.3	13.4
FCOS	96.5	99.2	89.8	95.7	91.3	81.5	92.4	28.5
Yang *et al.* (2021)	96.6	98.4	94.9	92.0	87.7	89.4	93.1	36.6
Liu *et al.* (2020)	96.0	99.5	92.8	92.3	90.1	90.5	93.5	36.5
YOLOv4	97.9	98.8	96.8	95.6	88.3	87.7	94.1	22.7
Proposed	98.0	98.7	97.2	92.7	91.8	91.6	95.0	21.6

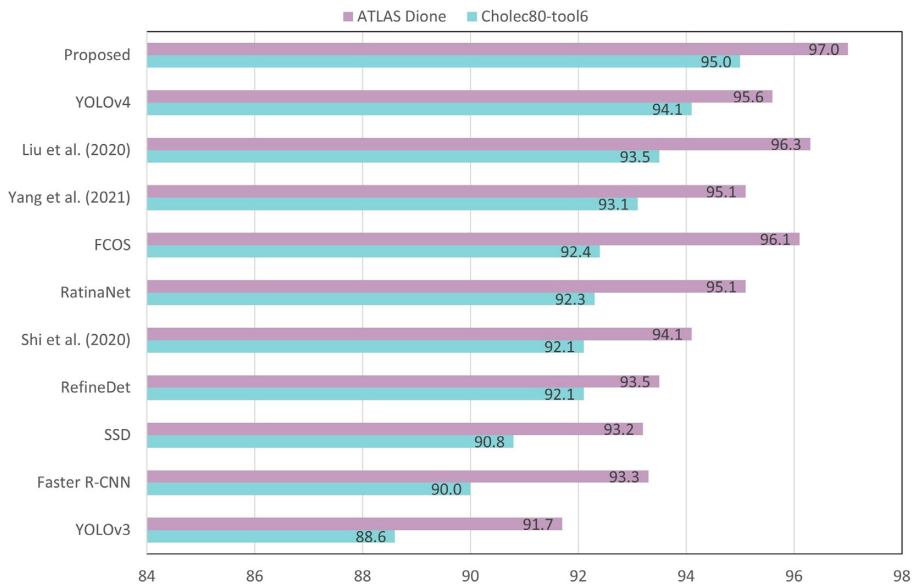

Figure 7: mAP for all methods applied to ATLAS Dione and Cholec80-tool6 datasets.

fore, the proposed method has better accuracy in datasets with complex content. These designs improve accuracy while reducing speed. In the ATLAS Dione and Cholec80-tool6 datasets, the speed of the method of Liu *et al.* (2020) was higher than that of the proposed method.

Similarly, the above analysis also applies to FCOS. If practical applications are considered, the method of Liu *et al.* (2020) and FCOS are suitable for simple surgical tasks, such as suturing and knotting. In simple surgical tasks, they are fast and accurate. The proposed method is more suitable for complex surgical tasks,

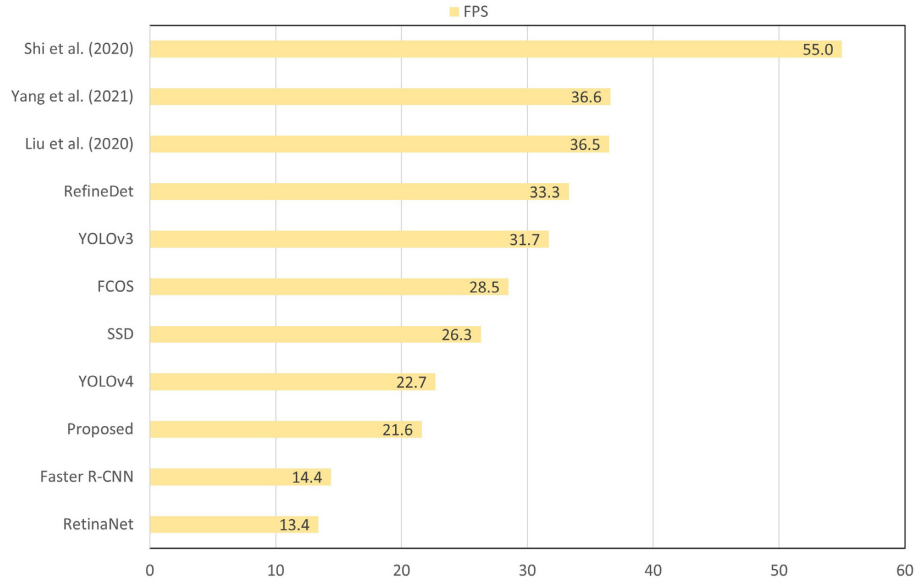

Figure 8: FPS for all methods applied to ATLAS Dione and Cholec80-tool6 datasets.

Table 5: Ablation test results for ATLAS Dione (mAP1) and Cholec80-tool6 (mAP2) datasets.

Method	mAP1 (%)	mAP2 (%)	FPS
Basic network	91.7	88.6	31.7
Basic + FENet20	93.9	90.3	24.2
Basic + EFFNet	95.1	92.4	27.8
Proposed	97.0	95.0	21.6

Table 6: Results for ATLAS Dione (mAP1) and Cholec80-tool6 (mAP2) datasets.

Method	mAP1 (%)	mAP2 (%)	FPS
Basic + EfficientNetV2	94.2	90.4	15.6
Basic + FENet20	93.9	90.3	24.2

such as a cholecystectomy. In complex surgical tasks, the proposed method can achieve better detection results.

The detection methods in comparison tests can also be divided into two-stage and one-stage methods. Faster R-CNN is a classical two-stage method. In tests, it did not meet the real-time standard and had a lower accuracy than the proposed method. YOLOv3 is a classical one-stage method. In tests, it was faster than the proposed method, but far less accurate.

The proposed method is a one-stage method, but its network architecture is more complex and has more layers. In particular, EFFNet performs two rounds of feature fusion to enhance the utilization of low-level and high-level feature information. These designs compensate for the low accuracy of the one-stage method, and although they also bring the problem of speed reduction, the proposed method still meets the real-time standard.

3.3. Ablation study

To more intuitively demonstrate the improvements brought about by FENet20 and EFFNet to the basic network, we performed ablation tests on the ATLAS Dionet and Cholec80-tool6 datasets. In Table 5, the Basic network is YOLOv3, the backbone of which is DarkNet53. When DarkNet53 was replaced with FENet20, the resulting Basic + FENet20 network had a higher accuracy than the original network (by 2.2% for mAP1 and by 1.7% for mAP2), but the speed dropped by 7.5 FPS. When the feature-fusion part of YOLOv3 was replaced with EFFNet, the resulting Basic + EFFNet network also had a higher accuracy than the original network (by 3.4% for mAP1 and by 3.8% for mAP2), but the speed dropped by 3.9 FPS.

DarkNet53 mainly consists of residual modules and FENet20 consists of Fused-MBconv and MBconv modules. Among these, the MBconv module contains the SE module (Hu *et al.*, 2020). In the process of extracting image features, the SE module made the effective feature-map weight heavier and the invalid or less effective feature-map weight smaller by adaptively giving the channel weights. As a result, the SE module extracted features more effectively than normal convolution. The results of the ablation tests showed that Basic + FENet20 network was more accurate than the Basic network. This also demonstrated that FENet20 had a better feature-extraction capability than DarkNet53.

Compared with the feature-fusion part of YOLOv3, EFFNet included two rounds of feature fusion. In the first round of feature fusion, the SPP module increased the receptive field and separated the most significant context features. The second round of feature fusion more comprehensively fused low- and high-level features. The results of the ablation tests showed that EFFNet not only had a small impact on speed, but also improved accuracy.

Overall, compared with the Basic network, the proposed method improved the accuracy by 5.3% (mAP1) and 6.4% (mAP2), but decreased the speed by 10.1 FPS. A series of ablation tests proved that FENet20 and EFFNet greatly improved accuracy. Although FENet20 and EFFNet reduced the speed, they still met the real-time standard.

To verify that FENet20 was more applicable to this study than EfficientNetV2, we added a comparative test. In Table 6, Basic + EfficientNetV2 corresponds to the network in which Dark-

Net53 was replaced with EfficientNetV2. The accuracy remained almost unchanged, but the speed was lower by 8.6 FPS than that obtained with Basic + FENet20. EfficientNetV2 is an excellent network, but it may be relatively bulky as a backbone. We optimized the number of layers of EfficientNetV2 in a reasonable way, with the aim of increasing the detection speed and meeting the real-time standard. Compared to EfficientNetV2, FENet20 was lighter and more efficient.

4. Discussion

Among all the detection methods in comparison tests, the proposed method not only has the highest accuracy, but also meets the real-time standard. The proposed method is both anchor-based and one-stage. Compared with the anchor-free methods [Liu *et al.* (2020) and FCOS], it exhibits better accuracy on datasets with complex content. Compared with two-stage Faster R-CNN and one-stage YOLOv3, it has the highest accuracy, but the speed is lower than that of YOLOv3.

The anchor box in the proposed method constrains the predicted object range and adds a priori experience of size, so as to realize multiscale learning. In addition, FENet20 and EFFNet make the proposed method's network architecture more complex and have more layers. Although these designs bring about the problem of speed reduction, the proposed method still meets the real-time standard.

In the ablation study, both FENet20 and EFFNet promoted the improvement of accuracy. FENet20 had a good feature-extraction capability. EFFNet performed two rounds of feature fusion to enhance the utilization of low- and high-level feature information. In the study of FENet20 and EfficientNetV2, we found that EfficientNetV2 was relatively bulky as the backbone. FENet20 optimized by EfficientNetV2 was lighter and more efficient.

However, several shortcomings are noteworthy. First, compared with the current, large datasets, the size of the Cholec80-tool6 dataset was small (4013 frames), and surgical tool detection still faced the problem of data shortage. Second, the proposed method met the real-time standard, but it was still slower than the speed-focused detection methods. Since speed is an important indicator of object detection method performance, the proposed method must be further improved.

Given the above shortcomings, we plan to continue to annotate surgical tools in the Cholec80 dataset and make a larger dataset in the near future. However, for other types of surgery, we still face the problem of data shortage. Considering the long-term development, the proposed method should be developed to be weakly supervised or even unsupervised, so as to avoid the fatal impact caused by data shortage. In object detection, the anchor-free concept has great development potential since it eliminates the amount of calculation brought about by using an anchor, which makes object detection methods execute more in real time and with high precision. The proposed method should be developed to function anchor-free to promote the improvement of speed.

The clinical significance of the proposed method is also worth discussing. Surgical video analysis allows an objective and valid assessment of the surgeons' surgical skills in carrying out surgical procedures and prevents complications due to poor individual or team performance (Jin *et al.*, 2018a). Real-time automated surgical video analysis relies on real-time motion information of the tool. The proposed method meets the real-time detection standard and provides a basic guarantee for real-time surgical video analysis. Furthermore, it achieves the best accuracy compared to other methods evaluated in this study, indicating that the proposed method can provide more accurate information about the position of the tool in surgical video analysis. The precise positioning of surgical tools provides a more objective basis for the assessment of the surgeons' surgical skills. Note that the output of surgical video analysis can only be used as a reference opinion, and the final decision is still up to the surgeon.

5. Conclusions

A method for real-time surgical tool detection and manual annotation of a new dataset, the Cholec80-tool6 dataset, is proposed in this study. The proposed method consists of FENet20, EFFNet, WF, and predictor. FENet20, as the backbone, was quite portable and efficient and improved the detection accuracy. EFFNet implemented two rounds of feature fusion to enhance the utilization of low- and high-level feature information, resulting in higher accuracy. These designs also bring about the problem of speed reduction, but the proposed method still meets the real-time standard. In the future, the proposed method should be developed to be weakly supervised or even unsupervised, so as to avoid the fatal impact caused by data shortage. We plan to study the anchor-free concept to reduce the amount of calculation brought about by using an anchor and thereby improve the method's speed.

Acknowledgments

This work was supported by China's National Key Research and Development Program under Grant No. 2019YFB1311300.

Conflict of Interest Statement

None declared.

References

Al Hajj, H., Lamard, M., Conze, P., Cochener, B., & Quellec, G. (2018). Monitoring tool usage in surgery videos using boosted convolutional and recurrent neural networks. *Medical Image Analysis*, 47, 203–218. https://doi.org/10.1016/j.media.2018.05.001.

Alsheakhali, M., Yigitsoy, M., Eslami, A., & Navab, N. (2015). Surgical tool detection and tracking in retinal microsurgery. In *Proceedings of SPIE*(pp. 245–250). SPIE. https://doi.org/10.1117/12.2082335.

Bochkovskiy, A, Wang, C. -Y., & Liao, H. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. https://doi.org/10.48550/arXiv.2004.10934.

Choi, B., Jo, K., Choi, S., & Choi, J. (2017). Surgical-tools detection based on convolutional neural network in laparoscopic robot-assisted surgery. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*(pp. 1756–1759). IEEE. https://doi.org/10.1109/EMBC.2017.8037183.

Du, X., Kurmann, T., Chang, P., Allan, M., Ourselin, S., Sznitman, R., Kelly, J. D., & Stoyanov, D. (2018). Articulated multi-instrument 2-D pose estimation using fully convolutional networks. *IEEE Transactions on Medical Imaging*, *37*(5), 1276–1287. https://doi.org/10.1109/TMI.2017.2787672.

Fried, M. P., Kleefield, J., Gopal, H., Reardon, E., Ho, B. T., & Kuhn, F. A. (1997). Image-guided endoscopic surgery: Results of ac-

curacy and performance in a multicenter clinical study using an electromagnetic tracking system. *The Laryngoscope*, 107(5), 594–601. https://doi.org/10.1097/00005537-199705000-00008.

Gao, C., Unberath, M., Taylor, R., & Armand, M. (2019). Localizing dexterous surgical tools in X-ray for image-based navigation. https://doi.org/10.48550/arXiv.1901.06672.

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 386–397. https ://doi.org/10.48550/arXiv.1703.06870.

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (GELUs). https://doi.org/10.48550/arXiv.1606.08415.

Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2020). Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8), 2011–2023. https://doi.org/10.1109/TPAMI.2019.2913372.

Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. Q. (2016). Deep networks with stochastic depth. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science*(Vol. 9908, pp. 646–661). Springer. https://doi.org/10.1007/978-3-319-46493-0_39.

Jin, A., Yeung, S., Jopling, J., Krause, J., Azagury, D., Milstein, A., & Fei-Fei, L. (2018a). Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 691–699. https://doi.org/10.1109/WACV.2018.00081.

Jin, Y., Dou, Q., Chen, H., Yu, L., Qin, J., Fu, C., & Heng, P. (2018b). SV-RCNet: Workflow recognition from surgical videos using recurrent convolutional network. *IEEE Transactions on Medical Imaging*, 37(5), 1114–1126. https://doi.org/10.1109/TMI.2017.2787657.

Jin, Y., Li, H., Dou, Q., Chen, H., Qin, J., Fu, C., & Heng, P. (2020). Multi-task recurrent convolutional network with correlation loss for surgical video analysis. *Medical Image Analysis*, 59, 101572–101572. https://doi.org/10.1016/j.media.2019.101572.

Joskowicz, L., Milgrom, C., Simkin, A., Tockus, L., & Yaniv, Z. (1998). FRACAS: A system for computer-aided image-guided long bone fracture surgery. *Computer Aided Surgery*, 3(6), 271–288. https://doi.org/10.1002/(SICI)1097-0150(1998)3:6(271::AID-IGS1)3.0.CO;2-Y.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. 2, 1097–1105. https://dl.acm.org/doi/10.1145/3065386.

Krupa, A., Gangloff, J., Doignon, C., de Mathelin, M. F., Morel, G., Leroy, J., Soler, L., & Marescaux, J. (2003). Autonomous 3-D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing. *IEEE Transactions on Robotics and Automation*, 19(5), 842–853. https://doi.org/10.1109/TRA.2003.817086.

Kurmann, T., Marquez Neila, P., Du, X., Fua, P., Stoyanov, D., Wolf, S., & Sznitman, R. (2017). Simultaneous recognition and pose estimation of instruments in minimally invasive surgery. In M. Descoteaux, L. Maier-Hein, A. Franz, P. Jannin, D. Collins, & Duchesne (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2017. MICCAI 2017. Lecture Notes in Computer Science*(Vol. 10434, pp. 505–513). Springer. https://doi.org/10.1007/978-3-319-66185-8_57.

Laina, I., Rieke, N., Rupprecht, C., Vizcaíno, J. P., Eslami, A., Tombari, F., & Navab, N. (2017). Concurrent segmentation and localization for tracking of surgical instruments. In M. Descoteaux, L. Maier-Hein, A. Franz, P. Jannin, D. Collins, & S. Duchesne (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2017. MICCAI 2017. Lecture Notes in Computer Science*(Vol. 10434, pp. 664–672). Springer. https://doi.org/10.1007/978-3-319-66185-8_75.

Lee, C., Wang, Y., Uecker, D. R., & Wang, Y. (1994). Image analysis for automated tracking in robot-assisted endoscopic surgery. In *Proceedings of 12th International Conference on Pattern Recognition*(pp. 88–92). IEEE. https://doi.org/10.1109/ICPR.1994.576232.

Lee, W., Roh, M., Lee, H., Ha, J., Cho, Y., Lee, S., & Son, N. (2021). Detection and tracking for the awareness of surroundings of a ship based on deep learning. *Journal of Computational Design and Engineering*, 8(5), 1407–1430. https://doi.org/10.1093/jcde/qwab053.

Lin, T., Goyal, P., Girshick, R., He, K., & Dollar, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 318–327. https://doi.org/10.1109/TPAMI.2018.2858826.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., & Berg, A. C. (2016). SSD: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science*(Vol. 9905, pp. 21–37). Springer. https://doi.org/10.1007/978-3-319-46448-0_2.

Liu, S., Huang, D., & Wang, Y. (2019). Learning spatial fusion for single-shot object detection. https://doi.org/10.48550/arXiv.1911.09516.

Liu, Y., Zhao, Z., Chang, F., & Hu, S. (2020). An anchor-free convolutional neural network for real-time surgical tool detection in robot-assisted surgery. *IEEE Access*, 8, 78193–78201. https://doi.org/10.1109/ACCESS.2020.2989807.

Mishra, K., Sathish, R., & Sheet, D. (2017). Learning latent temporal connectionism of deep residual visual abstractions for identifying surgical tools in laparoscopy procedures. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*(pp. 2233–2240). IEEE. https://doi.org/10.1109/CVPRW.2017.277.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science*(Vol. 9912, pp. 483–499). Springer. https://doi.org/10.1007/978-3-319-46484-8_29.

Ni, Z., Bian, G., Xie, X., Hou, Z., Zhou, X., & Zhou, Y. (2019). RASNet: Segmentation for tracking surgical instruments in surgical videos using refined attention segmentation network. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*(pp. 5735–5738). IEEE. https://doi.org/10.1109/EMBC.2019.8856495.

Nwoye, C. I., Mutter, D., Marescaux, J., & Padoy, N. (2019). Weakly supervised convolutional LSTM approach for tool tracking in laparoscopic videos. *International Journal for Computer Assisted Radiology and Surgery*, 14(6), 1059–1067. https://doi.org/10.1007/s11548-019-01958-6.

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

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*(pp. 6517–6525). IEEE. https://doi.org/10.1109/CVPR.2017.690.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. https://doi.org/10.48550/arXiv.1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*(pp. 658–666). IEEE. https://doi.org/10.1109/CVPR.2019.00075.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. Wells, & A. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science*(Vol. *9351*, pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28.

Sarikaya, D., Corso, J. J., & Guru, K. A. (2017). Detection and localization of robotic tools in robot-assisted surgery videos using deep neural networks for region proposal and detection. *IEEE Transactions on Medical Imaging*, *36*(7), 1542–1549. https://doi.org/10.1109/TMI.2017.2665671.

Shelhamer, E., Long, J., & Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(4), 640–651. https://doi.org/10.1109/TPAMI.2016.2572683.

Shi, P., Zhao, Z., Hu, S., & Chang, F. (2020). Real-time surgical tool detection in minimally invasive surgery based on attention-guided convolutional neural network. *IEEE Access*, *8*, 1–1. https://doi.org/10.1109/ACCESS.2020.3046258.

Stoyanov, D. (2012). Surgical vision. *Annals of Biomedical Engineering*, *40*(2), 332–345. https://doi.org/10.1007/s10439-011-0441-z.

Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller models and faster training. https://doi.org/10.48550/arXiv.2104.00298.

Tian, Z., Shen, C., Chen, H., & He, T. (2019). FCOS: Fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*(pp. 9626–9635). IEEE. https://doi.org/10.1109/ICCV.2019.00972.

Twinanda, A. P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M., & Padoy, N. (2017). EndoNet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Transactions on Medical Imaging*, *36*(1), 86–97. https://doi.org/10.1109/TMI.2016.2593957.

Vania, M., & Lee, D. (2021). Intervertebral disc instance segmentation using a multistage optimization mask-RCNN (MOM-RCNN). *Journal of Computational Design and Engineering*, *8*(4), 1023–1036. https://doi.org/10.1093/jcde/qwab030.

Vania, M., Mureja, D., & Lee, D. (2019). Automatic spine segmentation from CT images using convolutional neural network via redundant generation of class labels. *Journal of Computational Design and Engineering*, *6*(2), 224–232. https://doi.org/10.1016/j.jcde.2018.05.002.

Vardazaryan, A., Mutter, D., Marescaux, J., & Padoy, N. (2018). Weakly-supervised learning for tool localization in laparoscopic videos. In *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*(Vol. *11043*, pp. 169–179). Springer. https://doi.org/10.1007/978-3-030-01364-6_19.

Wang, S., Raju, A., & Huang, J. (2017). Deep learning based multi-label classification for surgical tool presence detection in laparoscopic videos. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*(pp. 620–623). IEEE. https://doi.org/10.1109/ISBI.2017.7950597.

Wang, G., Wang, K., & Lin, L. (2019a). Adaptively connected neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*(pp. 1781–1790). IEEE. https://doi.org/10.1109/CVPR.2019.00188.

Wang, S., Xu, Z., Yan, C., & Huang, J. (2019b). Graph convolutional nets for tool presence detection in surgical videos. In A. Chung, J. Gee, P. Yushkevich, & S. Bao (Eds.), *Information Processing in Medical Imaging. IPMI 2019. Lecture Notes in Computer Science*(Vol. *11492*, pp. 467–478). Springer. https://doi.org/10.1007/978-3-030-20351-1_36.

Yang, W., Hu, C., Meng, M. Q. -H., Song, S., & Dai, H. (2009). A six-dimensional magnetic localization algorithm for a rectangular magnet objective based on a particle swarm optimizer. *IEEE Transactions on Magnetics*, *45*(8), 3092–3099. https://doi.org/10.1109/TMAG.2009.2019116.

Yang, Y., Zhao, Z., Shi, P., & Hu, S. (2021). An efficient one-stage detector for real-time surgical tools detection in robot-assisted surgery. In B. W. Papież, M. Yaqub, J. Jiao, A. I. L. Namburete, & J. A. Noble (Eds.), *Medical Image Understanding and Analysis. MIUA 2021. Lecture Notes in Computer Science*(Vol. *12722*, pp. 18–29). Springer. https://doi.org/10.1007/978-3-030-80432-9_2.

Yengera, G., Mutter, D., Marescaux, J., & Padoy, N. (2018). Less is more: Surgical phase recognition with less annotations through self-supervised pre-training of CNN-LSTM networks. https://doi.org/10.48550/arXiv.1805.08569.

Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018). Single-shot refinement neural network for object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*(pp. 4203–4212). IEEE. https://doi.org/10.1109/CVPR.2018.00442.

Zhao, Z., Voros, S., Weng, Y., Chang, F., & Li, R. (2017). Tracking-by-detection of surgical instruments in minimally invasive surgery via the convolutional neural network deep learning-based method. *Computer Assisted Surgery*, *22*(sup1), 26–35. https://doi.org/10.1080/24699322.2017.1378777.

Zhao, Z., Cai, T., Chang, F., & Cheng, X. (2019a). Real-time surgical instrument detection in robot-assisted surgery using a convolutional neural network cascade. *Healthcare Technology Letters*, *6*(6), 275–279. https://doi.org/10.1049/htl.2019.0064.

Zhao, Z., Voros, S., Chen, Z., & Cheng, X. (2019b). Surgical tool tracking based on two CNNs: From coarse to fine. *Journal of Engineering*, *2019*(14), 467–472. https://doi.org/10.1049/joe.2018.9401.

Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. https://doi.org/10.48550/arXiv.1904.07850.