Knowledge-Based Systems 239 (2022) 107860

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A new weakly supervised strategy for surgical tool detection☆

Yao Xue [a], Siming Liu [a], Yonghui Li [a], Ping Wang [a], Xueming Qian [a,b,*]

[a] School of Information and Communication Engineering, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, 710049, China
[b] Ministry of Education Key Laboratory for Intelligent Networks and Network Security, China

ARTICLE INFO

Article history:
Received 16 June 2021
Received in revised form 28 September 2021
Accepted 2 December 2021
Available online 11 January 2022

Keywords:
Surgical tool detection
Weakly supervised
Surgical images

ABSTRACT

Surgical tool detection is a recently active research area. It is the foundation to a series of advanced surgical support functions, such as image guided surgical navigation, forming safety zone between surgical tools and sensitive tissues. Previous methods rely on two types of information: tool locating signals and vision features. Collecting tool locating signals requires additional hardware equipments. Vision based methods train their detection models using strong annotations (e.g. bounding boxes), which are quite rare and expensive to acquire in the field of surgical image understanding. In this paper, we propose a Pseudo Supervised surgical Tool detection (PSTD) framework, which performs explicit detection refinement by three levels of associated measures (pseudo bounding box generation, real box regression, weighted boxes fusion) in a weakly supervised manner. On the basis of PSTD, we develop a Bi-directional Adaption Weighting (BAW) mechanism in our tool classifier for contextual information mining by creating competition or cooperation relationships between channels. By only using image-level tool category labels, the proposed method yields state-of-the-art results with 87.0% mAP on a mainstream surgical image dataset: Cheloc80.

1. Introduction

Nowadays, Minimally Invasive Surgery (MIS) is a preferred technique for many surgery procedures, and is able to avoid many major drawbacks of open surgery, for example lengthening patient hospitalization and recovery time. MIS has experienced its recent development with the introduction of surgery assisted robot. With the help of robotic tools, surgeon hand movement and force can be converted into gentle scale in real time, so that sophisticated surgery procedures can be fulfilled with ease.

However, MIS suffers with the reduced view fields on the surgical site, which could affect visual understanding of surgeons and restrict the movement freedom of surgical tools. To facilitate accurate manipulation of tools on surgical sites, it is important to track the spatial relationship between anatomy areas and tools. Computer vision assisted intervention is a solution to a series of advanced surgical support functions, such as image guided surgical navigation [1], segmentation of organs in camera field of view [2], development of algorithms to form safety zone between surgical tool and sensitive tissues [3], and surgical tool detection [4,5], segmentation [6] and pose estimation [7]. Large-scale object classification and detection have seen the effectiveness of deep neural networks. In the medical image field, computer vision based object detection has also shown huge potentials by the introduction deep neural network, e.g. [4,8]. However in the field of Robot-Assisted Surgery (RAS) images, these advances are not yet fully explored.

In this paper, we propose a method using deep convolutional neural networks (CNNs) for understanding RAS images and fast detection of surgical tools. Literatures have investigated the task of surgical tool detection in distinct surgical fields, such as: retinal microsurgery [9,10], abdominal MIS [11,12]. Early solutions are based on markers on surgical tools [13] or active fiducials e.g. laser pointers. While in practice, such methods require hardware modifications, hence are more difficult to be widely used clinically. In addition, they still inherently suffer from unstable markers and from occlusions. Subsequent methods depend on classical machine learning models such as Random Forests [14] or probabilistic trackers [15]. Recently, EndoNet [5] is designed to carry out surgical phase recognition and tool presence detection in a multi-task manner, where one of its output layers is responsible to localize present tools. While, ToolNet [16] is

☆ This work is supported in part by the NSFC, China under Grant 62103317, 61772407, 61732008. This work is partly supported by China Postdoctoral Science Foundation under Grant 2021M702600. This work is partly supported by Natural Science Foundation of Shaanxi Province, China under Grant 2021JQ-058, and Beilin District Science and Technology Program, China GX2130. This work is partly supported by Pazhou Lab, Guangzhou, China.

* Correspondence to: Xi'an Jiaotong University, West Xianning Road #28, Xi'an, Post code: 710049, China.
E-mail addresses: xueyao@xjtu.edu.cn (Y. Xue), qianxm@mail.xjtu.edu.cn (X. Qian).

https://doi.org/10.1016/j.knosys.2021.107860
0950-7051/

a network trained in a single-task manner that solely performs the tool presence detection. Another recent tool detection work is [4], which applies a region proposal network and a multimodal two stream convolutional network to jointly recognize object and detection on a fusion of image and temporal motion cues.

Despite their enormous success in various computer vision tasks, the models training requires a vast number of strong annotations, such as instance level labels (bounding boxes or centroid points) and pixel level labels (segmentation mask). Several natural scene datasets are created, but models pre-trained on these large-scale datasets cannot perform well for surgical tool detection, due to the data consistence gap between source domain and target domain.

More importantly, surgery image datasets for public use are quite rare. A few public available datasets include JIGSAWS [17], m2cai16-workflow and m2cai16-tool datasets [5], but none of these datasets offer tool annotations. Instead, they only provide whole image level labels that indicate which kind of tools are present or which surgical operation phase the current frame belongs to. This fact highlights the necessity and significance of Weakly Supervised surgical Tool Detection (WSTD) approach, which relies on image-level category labels that are easy and cheap to acquire.

Paradigm shift from WSTD to PSTD

Recent WSTD approaches [18–20] often utilize class activation maps [21] to refine their detection performance. These class activation map based methods impose thresholding on convolution feature maps for further regions of interest proposal. But the parameter in thresholding is tricky and hard to determine.

Other WSTD methods perform classification and localization jointly. But in many cases, the optimization objectives of the two sub-tasks are inconsistent. HaS [22] and ADL [20] have observed that localization is not compatible with classification if a single CNN model is used. Classification models aim to recognize the whole object, while localization models often pay attention on the most discriminative parts of the object. In comparison, we choose to factorize surgical tools detection into two independent sub-tasks: the class-agnostic bounding box regression and the tool classification.

In this paper, we propose a surgical tool detection scheme in a rather practical way, where instead of relying on class activation maps, we make a paradigm shift from WSTD to Pseudo Supervised surgical Tool Detection (PSTD). In the PSTD framework, we firstly design a pseudo bounding box generation scheme, which provides us with initial tool detection information in a weakly supervised manner with low computational cost. For this box generation scheme, we build a set of green-background reference images for each tool category. The reference images of a tool category contain abundant visual appearances of the category with good resolution to discriminative details and common parts shared by distinct viewpoints. So that the quality of generated pseudo bounding boxes can be guaranteed in the first level. After that, a bounding box regressor is trained to refine pseudo bounding boxes. To further improve detection accuracy, we apply Weighted Mean Boxes Fusion (WMBF) strategy to fuse the redundant output boxes from the regressor.

Introduction of Bi-directional Channel Adaption

In addition to PSTD, we propose a Bi-directional (competition or cooperation) Adaption Weighting (BAW) mechanism into our surgical tool classifier. Since the distance and viewpoint of camera bring huge variation to the appearance of surgical tools. Recent object detection works [23,24] have suggested that the contributions of different convolution channels are not fixed, but modulated by current input and stimulus. Local Response Normalization (LRN) benefits from introducing only competition relationship among neurons. SEnet [23] uses global information to adaptively emphasize informative channels that have proper convolution activation maps.

In comparison, BAW is able to create two types of relationship (competition and cooperation) among different channels during the training process. By combining normalization with a bi-directional gating operation, the contribution of each channel can be enhanced or suppressed. When the gating weight of one channel is activated positively, BAW promotes this channel to compete with other channels as in LRN. When the gating weight is activated negatively, BAW encourages this channel to cooperate with the others. Furthermore in SEnet, two fully-connected (FC) layers are leveraged to compute a set of weights for different channels. The FC layers have the parameter complexity of $O(Ch^2)$ (Ch is the number of channels). While, BAW does not employ FC operations and has a smaller parameter complexity $O(Ch)$.

The *contributions* of this paper are summarized as.

(1) We propose a surgical tool detection scheme that only utilizes image level tool category labels instead of requiring strong tool annotations which are rare and expensive to acquire in the community of robot-assisted surgery.

(2) We create a pseudo supervised surgical tool detection framework, which consists of four associated modules: pseudo bounding box generation, box regressor, weighted mean boxes fusion and a tool classifier with bi-directional channel adaption capacity.

(3) To deal with tool appearance variation issue, we develop a Bi-directional (competition and cooperation) Adaption Weighting (BAW) mechanism, to adaptively emphasize informative channels and suppress less useful ones.

(4) During pseudo bounding box generation, we design a $1 + N$ mode (1 input image and N reference images of the same category) to purify the pseudo bounding boxes. Within the $1 + N$ mode, input image areas that are similar to or simultaneously shared by reference images will be selected as region proposal.

2. Related work

In early years, most surgery tool detection approaches tend to simplify the detection task into an image color segmentation or thresholding. Color markers, color coding tools and laser projectors were commonly used at that time. For example, [13] is based on a barcode marker. Fan et al. [25] develop a 3D-marker based spatial position estimation system for surgical tool navigation. Du et al. [11] develop a 2D tracker, which is built on a SIFT-based generalized hough transform, and use it to initialize a 3D tracker for each frame. But these approaches share a common drawback that they all need additional hardware or manufacturing, which set great limits on their applications.

Recently, pure vision-based approaches are proposed. Sznitman et al. [15] train a part-based multi-class classifier and then use sliding window to localize tools. Although they propose an early-stop algorithm, the time cost still needs to be reduced. Another visual method is shape matching. Bouget et al. [26] present a two-stage method for joint tool detection and pose estimation. Their results indicate that performing semantic labeling as intermediate task can improve detection performance. Colleoni et al. [27] introduce three-dimensional convolutional layers into a encoder–decoder architecture to jointly extract spatio-temporal information, which proves to be useful when dealing with training images with unseen backgrounds. Bouget et al. [28] turn the tool presence detection problem into a multi-label classification problem. In order to localize tools, they have to traverse the whole image pixel by pixel.

DPM [29] proposes a Deformable Parts Model, which effectively captures both the occlusion and articulation information which proves to be successful, but the detecting speed is slow somehow. EndoNet [5] performs surgical phase recognition and tool presence detection in a multi-task manner and claims that the multi-task manner does not compromise its performance in detecting the tools. Sarikaya et al. [4] utilize a Region Proposal Network to fuse the information of two CNN processing streams of two modalities and feed the output to a Fast RCNN to train a surgical tool detector. It outperforms Faster-RCNN with much more time cost. One tool detection work [30] models a surgical instrument as an articulated object, and develops a gradient-based pose estimation infers the location of the instrument parts.

Quite recently, deep neural networks present their superior potentials in various tasks, such as crowd counting [31], object segmentation [32], face detection [33]. Several cutting-edge general object detectors are also proposed for surgical tool detection. FCOS [34] is a fully convolutional one-stage object detector for object detection. FCOS is anchor box free by eliminating the predefined set of anchor boxes. RefineDet [35] proposes a novel single-shot based detector, which is believed to simultaneously maintain high accuracy of two-stage approaches and high efficiency of one-stage approaches. ATSS [36] points out that the essential difference between anchor-based and anchor-free detection is actually how to define positive and negative training samples.

As these approaches belongs to strong supervision, they need pixel-level bounding box annotation for model training. But for surgery tool detection task, the data in this domain is rare and hard to collect, let along pixel-level bounding box annotations. In order to detect objects with only image-level annotation, some approaches that base on multiple instance learning (MIL) framework have been proposed. Bilen et al. [37] build a weakly supervised deep detection network to perform object localization and classification at the same time. Tang et al. [38] improve the weakly supervised object detection performance using online instance classifier refinement (OICR), but it is easily trapped into local optima. Gao et al. [39] introduce a count-based region selection algorithm into OCIR to boost performance. However, it also requires additional labor to make count annotation. Cheng et al. [40] adopt object instance mining framework to address the problem of missing object instances and make the approach more robust for local optima to some extent. Slightly similar to our work, [41] belongs to pseudo supervised learning methods and also has pseudo label generation module. But [41] lacks the mechanisms (e.g. $1 + N$ mode used during pseudo bounding box generation, the competition capacity of our BAW) to filter out interferes from pseudo bounding boxes, which are not that accurate by nature.

3. The proposed method

Fig. 1 presents the overall framework of our approach. Pseudo bounding box generation, real bounding box regression and tool category classifier are discussed in Sections 3.1, 3.2 and 3.3 respectively.

3.1. Pseudo bounding box generation

Pseudo bounding box generation is the key factor that distinguishes PSTD from WSTD. Detection methods are naturally a solution for bounding box generation, because they can directly predict bounding boxes and classification results at the same time. While weakly-supervised or co-supervised methods that provide noisy bounding boxes can also generate good results on detection tasks. In comparison to detection methods, weakly supervised methods, e.g. DDT [42] have both good performance and low computational cost. Here we choose DDT to generate pseudo bounding box (see Table 1). Suppose R is a training image sub-set which contains n images sharing the same label. For each image $I \in R$, its feature map G_i can be generated using a pre-trained model F. Gather all the feature maps of the n images in a set G, then we can obtain the eigenvector P by applying principal component analysis on G. Using P as weight vector, a channel-level weighed sum on each G_i can provide us the heat map H_i for each image I. Upsample H_i to original size, then we get the pseudo bounding box by employing zero thresholding and max connected component analysis.

Table 1
List of notations used in this section.

Notation	Definition
R	a training image sub-set containing images of the same category
N	the number of training images belonging to the same category
G_i	the feature map of ith image generated by a pre-trained model.
P	the eigenvector of G given by principal component analysis
H_i	the heatmap by channel-level weighed sum on each G_i
S_c	green-background reference image set of category c
x_{cj}	jth training image of category c
b_{cj}	pseudo bounding boxes of image x_{cj}
Ch	the number of channels
NA	the number of anchors

However, directly using DDT in our surgery tool detection task will end in vain because the most common object in all surgical images is human tissue. It means the generated bounding boxes will mainly localize the backgrounds rather than surgery tools. In order to let DDT focus on tools, we make a 10 green-background reference image set $S_c = \{s_{c1}, s_{c2}, \dots, s_{c10}\}$ for each tool category i. Fig. 2 illustrates reference images of category *Grasper*. Please note that these class-corresponding reference sets are only available during pseudo bounding box generation for training images of each category.

For each training image x_{cj} belonging to category c, we combine it with its corresponding reference set S_c to form a package $P_j = \left\{x_{cj}, s_{c1}, s_{c2}, \dots, s_{c10}\right\}$. In this way, every training image x_{cj} is accompanied by 10 green-background reference images. All the 11 images contain a tool of the same category. Then we apply principal component analysis by DDT on P_j on the 11 images. And a set of heat maps will be generated for each image in P_j, denoted as $HP_j = \left\{hx_{cj}, hs_{c1}, hs_{c2}, \dots, hs_{c10}\right\}$, where hx_{cj} is the heat map of image x_{ij}. By thresholding heat map hx_{ij}, we can get the pseudo bounding boxes of x_{ij}, named as b_{ij}. Combine all pseudo bounding boxes of every frame, we obtain the pseudo bounding box set $B = \left\{B|B_c = b_{ij}, j = 1, \dots, m, i = 1, \dots, n\right\}$. Thus, every training image gets its tool bounding boxes, which focus more on the regions where surgical tools are present. We call the way that pseudo bounding box generation works as $1+N$ mode of pseudo bounding box generation. With the assistance of green background reference images for each tool category, the pseudo bounding boxes generated under $1 + N$ mode are more reliable than expectation.

3.2. Real bounding box regression

On the basis of pseudo bounding boxes generation, we train a bounding box regressor to further refine the pseudo bounding boxes generated. We extract frames from 80 videos in Cholec80 at a speed of 25 fps through ffmpeg. We use training image set X as input and the pseudo bounding box set B as annotation to train the box regressor. As our regressor use the bounding boxes generated by DDT as training data annotations, and DDT generates boxes with noise in nature. Consequently, our regressor will also predict redundant output, where a number of bounding boxes with various sizes and different centroid positions are generated around a single tool. In order to reduce the effect of

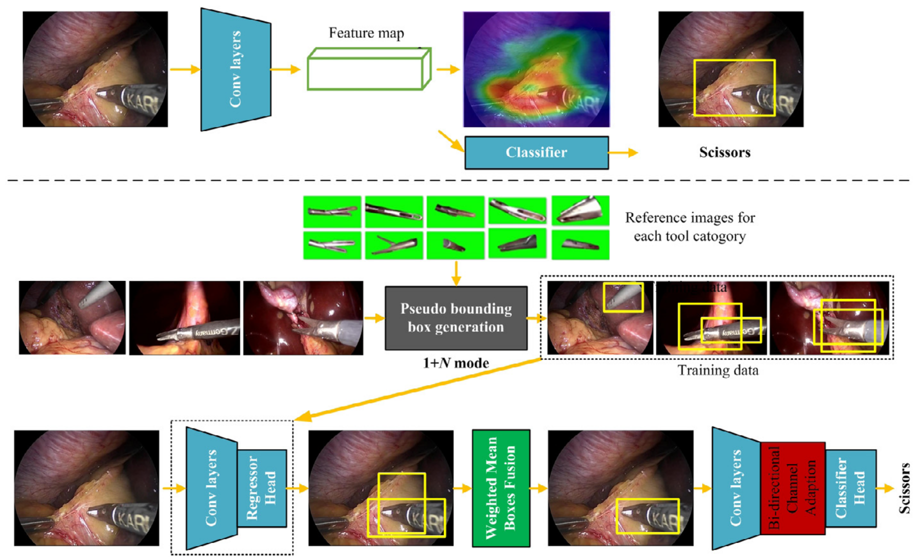

Fig. 1. System overview of the proposed method. Top row: traditional weakly supervised object detection; bottom row: our proposed Pseudo Supervised surgical Tool Detection (PSTD) scheme. The input to pseudo bounding box generation follows a $1+N$ mode (1 input image and N reference images belonging to the same category with the input image).

Fig. 2. Illustration of 10 green-background reference images for the tool category Grasper. The reference images of a tool category contain abundant visual appearances of the category with good resolution to discriminative details and common parts shared by distinct poses. So that the quality of generated pseudo bounding boxes can be guaranteed in the first level.

noise, we apply Weighted Mean Boxes Fusion (WMBF) strategy to fuse the redundant output boxes from the regressor.

A bounding box can be represented as $b_j = \{x_{j1}, y_{j1}, x_{j2}, y_{j2}\}$, where j indicates the index of this box, (x_1, y_1) and (x_2, y_2) are the coordinates of the top-left and bottom-right corner of the box respectively. Assume the bounding box set of tool category i in an image is $B_i = \{b_{i1}, \dots, b_{ij}, \dots, b_{im}\}$, where m is the number of boxes labeled category i in the image. The score set for all bounding boxes of B_i is $S_i = \{s_{i1}, \dots, s_{ij}, \dots, s_{im}\}$, where s_{ij} is the score for bounding box b_{ij}. Then we can calculate the weight w_{ij} for box b_{ij} using the following formula:

$$w_{ij} = \frac{s_{ij}}{\sum_{j=1}^{m} s_{ij}} \tag{1}$$

where m is the number of boxes labeled category i in the image. Here we simply use the confident score as the score for each bounding box. Finally, the output bounding box b_{class_i} for this tool of category i is expressed as:

$$b_{class_i} = \left\{ \sum_{j=1}^{m} w_{ij}x_{j1}, \sum_{j=1}^{m} w_{ij}y_{j1}, \sum_{j=1}^{m} w_{ij}x_{j2}, \sum_{j=1}^{m} w_{ij}y_{j2} \right\} \tag{2}$$

In this way, WMBF module generates fused boxes. So that, the coordinates of an output box will be adjusted jointly by both the coordinate and the confident scores of the input boxes.

3.3. Tool classifier

According to image-level labels of Cholec80, we extract the frames that share the same tool into a group. In many cases, tools take up a small space in a frame. Consequently, commonly used pre-processing method: resize and random crop cannot work well. Initial experiments show that only about 10% of random cropped image patches contain a complete surgical tool. So we manually crop 224 × 224 square image patches which contain a specific complete tool for training the classifier. We finally obtain a 8-classes dataset which consists of 3219 patches for each of the 7 classes and 7000 patches for background, totally 29,533 patches. We will release these manually cropped surgical tool images (or seen as manual annotated bounding boxes for Cholec80) as a contribution of this work for tool detection upon publication. Examples of processed data are shown in Fig. 3.

We use the 29,533 patches to train a 8-classes (7 tool classes and 1 background class) tool classifier. Our classifier is based on Resnet50 with Bi-directional Adaption Weighting (BAW) module, using cross entropy as its loss function. We initialize the Resnet50 layers and BAW layers with parameters pretrained from ImageNet-1k and random respectively, then finetune it on the 8-classes dataset.

3.3.1. Bi-directional adaption weighting

To deal with huge appearance variation of tools due to scale, distance, viewpoint etc., within our tool classifier, we propose a Bi-directional (competition or cooperation) Adaption Weighting (BAW) mechanism for adaptive channel-wise contextual information mining. Compared with SEnet, BAW is able to create competition or cooperation relationships among channels with smaller parameter complexity. Fig. 4 illustrates the structure of BAW module, which consists of three operations.

To make BAW learnable, we design a global context embedding operator, which embeds the global context and controls the weight for each channel before the normalization and a bi-directional gating adaptation operator, which adjusts the input feature channel-wisely. Let $Z \in R^{H \times W \times C}$ be an activation feature in a convolutional network, where H and W are the spatial height and width, and C is the number of channels. In general, BAW performs the following three operations.

Global Context Embedding

Because each learned filter operates with a local receptive field and consequently can only exploit contextual information within its receptive field. In order to have a large receptive field for exploiting channel dependencies, we firstly design a

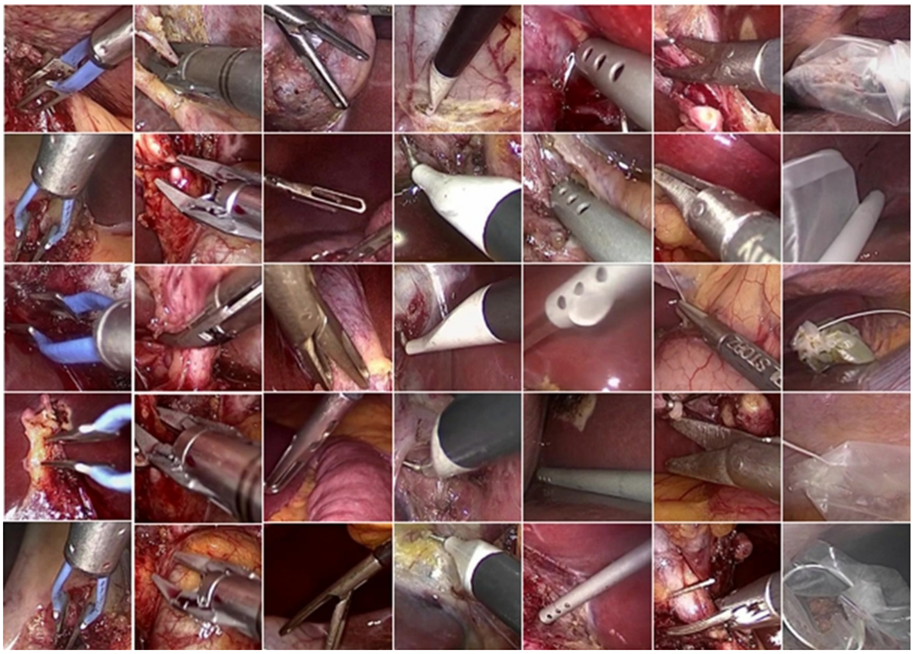

Fig. 3. Illustration of our surgical tool classifier training data.

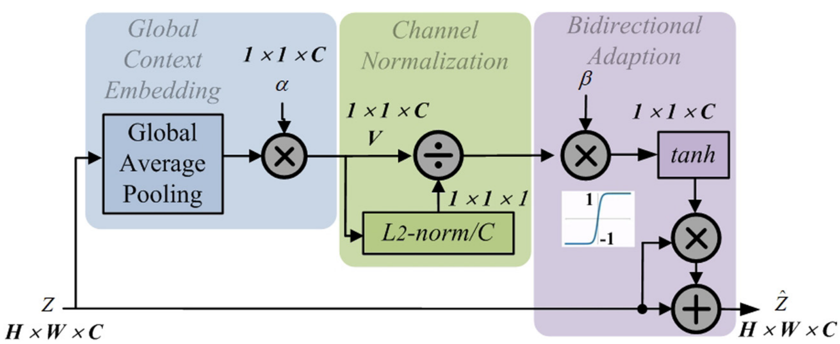

Fig. 4. Illustration of Bi-directional Adaption Weighting (BAW) block.

global context embedding module to aggregate global context information in each channel across their spatial dimensions. To do that, similar to SEnet [23], we use global average pooling to generate a channel-wise descriptor. But we add a trainable parameter: embedding weight α, which is responsible for adapting the embedding outputs. Thus the cth channel v_c is generated by shrinking input feature z_c through its spatial dimensions $H \times W$, with the embedding weight by:

$$v_c = F(z_c) = \frac{\alpha_c}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} z_c(i, j) \tag{3}$$

Channel Normalization

The second operation is channel normalization, which normalizes the original features with respect to the number of channels, with light-weight computing resource consumption. Similar to LRN, we use a L_2 normalization to operate across channels. Let $V = [v_1; \dots; v_C]$ be the input to channel normalization operation. The formula for the cth channel normalization is:

$$\hat{v_c} = \frac{C v_c}{\|V\|_2} = \frac{C v_c}{\sqrt{\sum_{c=1}^{C} v_c^2 + \epsilon}} \tag{4}$$

where ϵ is a small constant. The scalar C is used to normalize the scale of $\hat{v_c}$, avoiding a too small scale of $\hat{v_c}$ when C is large. Compared with SEnet that deploys two FC layers and has the parameter complexity of $O(C^2)$, BAW does not employ FC operations and has a smaller parameter complexity $O(C)$.

Bi-directional Adaption

To fully use the channel-wise dependencies captured by the first two operations, here we employ a bi-directional adaption mechanism. The previous channel normalization operation is parameter-free. Here we have a trainable parameter: gating weight β for learning to control the activation of gate channel-wisely. LRN benefits from creating only competitions among neurons. However, by introducing the bi-directional adaptation mechanism, the BAW can facilitate both competition and cooperation during the training process. Let the gating weight $\beta = [\beta_1, \dots, \beta_C]$, we design the following gating function: $\hat{z_c} = z_c[1 + tanh(\beta_c \hat{v_c})]$.

The scale of each input channel $\hat{v_c}$ will be strengthened or weakened by its corresponding gate, i.e., $1 + tanh(\beta_c \hat{v_c})$. When the gating weight of one channel β_c is activated positively, BAW promotes this channel to compete with the others as in LRN. When the gating weight is activated negatively, BAW encourages this channel to cooperate with the others. In this way, BAW is able to model both competition and cooperation among different channels by combining normalization methods and gating mechanisms.

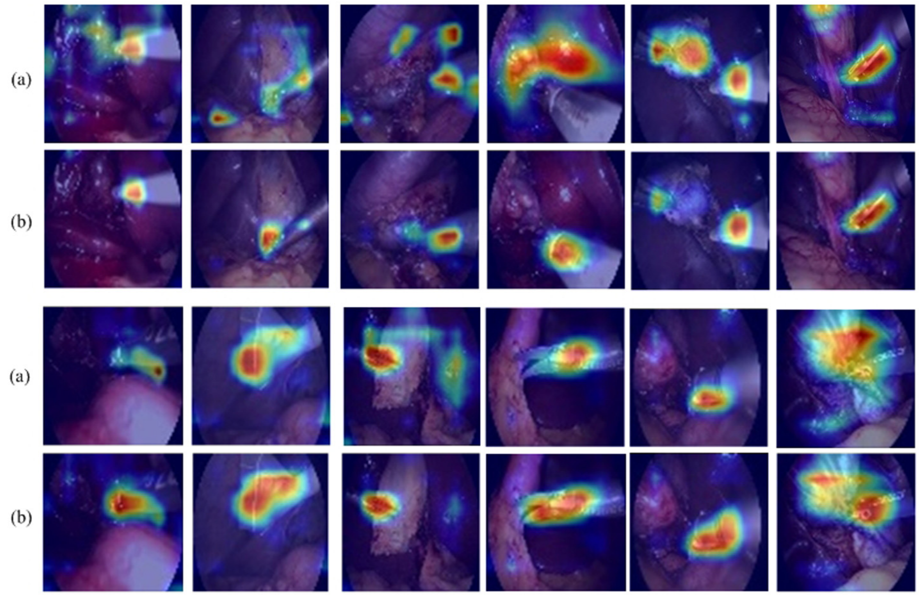

Fig. 5. Comparison of ordinary class activation maps (a) and feature maps (b) learned by Bi-directional Adaption Weighting (BAW).

Fig. 5 shows the channel heat maps comparison without and with the Bi-directional Adaption Weighting (BAW) module. The "a" rows visualize the class activation maps learned by ResNet-50 [43]. The "b" rows illustrate feature maps obtained from ResNet-50 as backbone followed by the proposed BAW module. All the networks are trained on our surgical tool classifier training data. From the heatmaps, one can observe pure ResNet fails to capture the discriminative regions. In comparison, higher scores are fired close to the regions where surgical tools are present, while most non-target regions have been suppressed. This heavily relies on our BAW module which benefits from adaptively capturing rich context information.

3.4. Classifier head and regressor head

The classifier head predicts the probability of object presence at each spatial position for C object classes. The design of this classifier is simple, see Fig. 6. Taking an input feature map with K channels from previous conv layers, the classifier head applies four 3×3 conv layers, each with K filters and each followed by ReLU activation, followed by a 3×3 conv layer with C filters. Finally sigmoid activations are attached to output the C binary predictions per spatial location. We use $K = 256$ as the default setting. In contrast to classic region proposal network, this classifier head uses only 3×3 convs, and does not share parameters with the box regression network.

The design of the regressor head is identical to the classier except that it terminates in $4 \times NA$ (number of anchors) linear outputs per spatial location, see Fig. 6. For each spatial location, these 4 outputs predict the relative offset between the predicted corner coordinates and the ground truth. Unlike most recent work, we use a class-agnostic bounding box regressor which uses fewer parameters and we found to be equally effective. The classifier and the regressor head, though sharing a common structure, use separate parameters.

4. Experiment

4.1. Dataset & evaluation metric

We use Cholec80 as our training and testing datasets. It contains 80 videos of cholecystectomy surgeries performed with 7 kinds of surgery tools. We use mean Average Precision (mAP) and Mean of intersection of union (mIoU) to evaluate our model. When calculating mAP, we consider an object positive, only if it satisfies the following two requirements: (1) the intersection over union (IoU) between its ground truth and predicted bounding box is bigger than 0.5; (2) the predicted class is the same as the ground truth class. mIoU is firstly used as an evaluation metric for semantic segmentation, but has been commonly used for evaluating detection performance [44]. To measure the computational complexity of models, we use three metrics: (1) number of total parameters, (2) number of MACs (Multiply-accumulate operations), (3) the average forward inference speed.

4.2. Implementation details

We build our model on PyTorch framework with the assistance of Nvidia GeForce GTX 1080 GPU.

4.2.1. Training bounding box regressor

We generate 10 green-background reference images for each kind of tool using Photoshop CC 2017. By pseudo bounding box generation, we get 13206 images with their pseudo bounding boxes as the training data for the regressor. The hyperparameters for the regressor are as follows: batch size 1, weight decay 0.0005, momentum 0.9. We set the start learning rate at 1e−4 and divide it by 10 every 10 epochs. The maximum epoch num is 30.

4.2.2. Training classifier

We have 29,533 images (224*224 sized), where 3219 images for each tool category and 7000 for background. We divide the dataset into training set and validation set at a proportion of 8:2. The hyperparameters for the classifier are as follows: batch size 4, weight decay 0.0005, momentum 0.9. We set the start learning rate at 0.001 and divide it by 10 every 10 epochs.

4.3. Comparison with state-of-the-art

We compare our proposed approach with 8 state-of-the-art methods including FCOS [34], RefineDet [35], RetinaNet [45], ToolNet [16], Faster-RCNN [46], Deformable Parts Model (DPM) [47], EndoNet [5] and ATSS [36]. There are both fully supervised

Classier
conv-1-s1 3×3×K
ReLu activation
conv-1-s1 3×3×K
ReLu activation
conv-1-s1 3×3×K
ReLu activation
conv-1-s1 3×3×K
ReLu activation
conv-1-s1 3×3×C
Sigmoid activation

Regression Network
conv-1-s1 3×3×K
ReLu activation
conv-1-s1 3×3×K
ReLu activation
conv-1-s1 3×3×K
ReLu activation
conv-1-s1 3×3×K
ReLu activation
conv-1-s1 3×3×C
Sigmoid activation
Fully-connected 1×1×(4*NA)

Fig. 6. Configuration of the classifier head and regressor head. All convolutional layers use padding=1, stride=1. Convolution layer parameters are denoted as "conv-(dilation rate)-stride kernel×kernel×filters".

Table 2
Average precision (AP) for all tools computed on the evaluation dataset of Cholec80. Red and Blue colors label the best and second best results in the mAP column. The PSTD-Net uses the complete configuration: Regressor+ClassiferWithBAW+WMBF.

Method	Bipolar	Clipper	Grasper	Hook	Irrigator	Scissors	Spec. bag	mAP
DPM [47]	70.6	68.4	82.3	73.4	67.5	73.4	69.0	70.7
Faster-RCNN [46]	83.1	80.5	79.6	79.2	81.0	78.2	81.2	80.4
ToolNet [16]	85.9	79.8	84.7	85.5	73.0	60.9	86.3	79.4
FCOS [34]	86.9	80.1	84.8	95.6	74.4	58.6	86.8	81.0
RefineDet [35]	83.5	82.9	87.1	85.7	84.3	84.6	87.9	85.0
RetinaNet [45]	80.1	79.0	79.6	81.3	82.6	79.6	83.0	80.7
ATSS [36]	84.2	80.6	80.9	87.3	81.6	83.5	85.7	83.4
EndoNet [5]	77.0	77.8	78.0	81.5	79.7	82.9	81.4	79.8
PSTD-Net	92.1	80.2	89.1	87.1	85.6	84.2	86.1	87.0

methods and weakly supervised methods among the comparison methods. Table 2 presents performances of all the methods on 7 tool classes with indication of supervision pattern of a method.

It can be seen that PSTD-Net obtains a significant edge over methods like DPM, ToolNet and EndoNet. FCOS, RefineDet and ATSS are 3 most competitive methods. RefineDet is a one-stage network and achieves better accuracy than two-stage methods like Faster-RCNN. FCOS is a fully convolutional one-stage object detector in a per-pixel prediction fashion. Unlike RetinaNet that relies on pre-defined anchor boxes, FCOS is anchor box free, as well as proposal free. With much simpler and flexible detection framework, FCOS achieves 5% higher mAP than RetinaNet. ATSS adopts an adaptive training sample selection to automatically select positive and negative samples according to statistical characteristics of objects. It significantly improves the performance of anchor-based and anchor-free detectors and bridges the gap between them. RetinaNet proposes to address this class imbalance by reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples.

In comparison, PSTD-Net achieves 87.0% mAP for detection of seven tools, with 2%, 3.6% and 5% improvement over RefineDet, ATSS and FCOS. EndoNet finds that tool presence detection can be done successfully without any explicit localization pre-processing steps (e.g. segmentation and ROI selection). In comparison, we still find that explicit region proposal refinement is beneficial for performance. In PSTD-Net, the region proposal refinement is done by three levels: pseudo bounding box generation, bounding box regression and bounding boxes fusion. This partly explains the improvement of PSTD-Net over RefineDet and EndoNet. EndoNet gives mAP at 79.8% as a weakly supervised method. The other factor comes from Bi-directional Adaption Weighting, which no longer treats the contribution of each channel equally.

Finally, the success of PSTD-Net and EndoNet indicates that weakly supervised mechanism is competent to localize tools with only relying on tools' image category labels. This is quite encouraging for practical use, since annotating a huge number of surgery images with image level labels is far more convenient and cheaper than providing bounding boxes.

Table 3 reports the performance comparison result measured by mIoU. The 9 comparison methods show quite similar performance ranking when measured by mAP and measured by mIoU. RefineDet, ATSS and FCOS are still the 3 most competitive methods. It can be seen that PSTD-Net achieves obtains 0.855 mIoU on the evaluation set, surpassing the second best model RefineDet by +0.02 mIoU (0.855 vs. 0.835). PSTD-Net is also +0.033 mIoU higher than ATSS, and +0.056 mIoU higher than FCOS.

Fig. 7 presents the results of our tool detection method on Cholec80 testing set. Red boxes indicate prediction boxes, green boxed indicate the ground-truth boxes. Each predicted box is associated with a category label and a confidence score in the range of [0, 1]. One can observe that our detection results present a wide range of scales and aspect ratios, and most results are close to the ground-truth in terms of positions and sizes. This explicitly demonstrates the effectiveness of the weakly supervised scheme in surgical tool detection. The following section gives further experiment analysis to PSTD-Net.

4.4. Computational complexity

To measure the computational complexity of models, we use three metrics: (1) number of parameters, (2) number of MACs, (3) average forward inference speed. We compare with 3 most competitive state-of-the-art methods: RefineDet, ATSS and FCOS.

Table 3
Performance measured by mean of intersection of union (mIoU). The configurations of all comparison methods are the same with Table 2.

Method	Bipolar	Clipper	Grasper	Hook	Irrigator	Scissors	Spec. bag	mIoU
DPM [47]	0. 601	0.719	0.816	0.748	0.694	0.719	0.670	0.728
Faster-RCNN [46]	0.782	0.796	0.817	0.694	0.840	0.794	0.819	0.792
ToolNet [16]	0.810	0.768	0.877	0.880	0.723	0.634	0.807	0.786
FCOS [34]	0.816	0.782	0.735	0.873	0.835	0.752	0.837	0.804
RefineDet [35]	0.842	0.824	0.836	0.839	0.845	0.834	0.825	0.835
RetinaNet [45]	0.849	0.823	0.804	0.855	0.810	0.616	0.835	0.799
ATSS [36]	0.850	0.817	0.803	0.825	0.803	0.830	0.825	0.822
EndoNet [5]	0.761	0.761	0.715	0.730	0.790	0.824	0.798	0.768
PSTD-Net	0.918	0.793	0.865	0.849	0.849	0.836	0.875	0.855

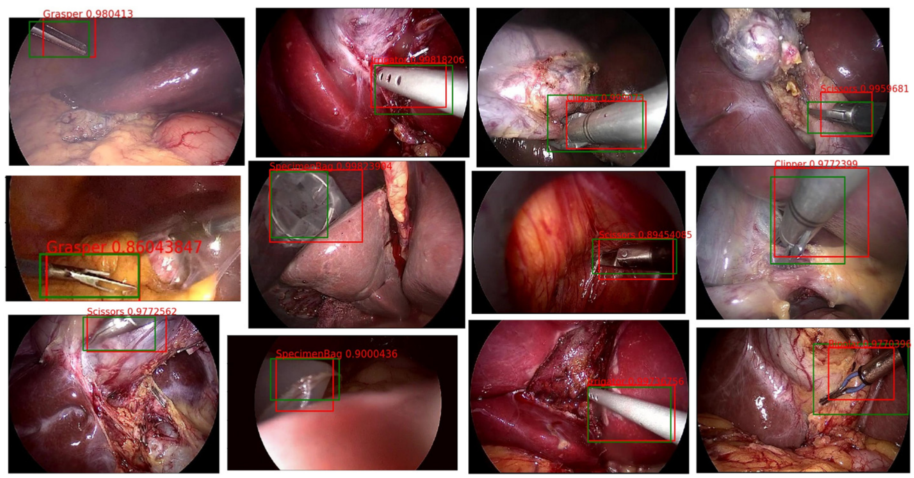

Fig. 7. Examples of our tool detection results on Cholec80 testing set. Our method detects objects of a wide range of scales and aspect ratios. Each output box is associated with a category label and a confidence score in [0, 1]. A score threshold of 0.86 is used to display these images. The running time for obtaining these results is 169 ms per image. Red box: detection, green box: ground-truth.

Table 4
Complexity comparison with 3 most competitive methods.

Method	Parameters	MAC	Speed	Supervision
RefineDet [35]	16.26 M	27.11 G	24.3180 ms	strong
ATSS [36]	127.95 K	1.40 G	9.3802 ms	strong
FCOS [5]	15.66 M	21.19 G	18.4170 ms	strong
PSTD-Net	5.83 M	3.15 G	15.2651 ms	weak

Table 4 summarizes the computational complexity comparison results. The "Supervision" column indicates whether a method works under full or weak supervision manner.

RefineDet is a one-stage network with comparable efficiency, its main computational complexity comes from its two interconnected modules. The use of feature alignment convolution can regress more accurate object locations, but requires a larger computational power and slows down inference speed. FCOS is another one-stage object detector. By eliminating the anchor boxes, FCOS completely avoids the complicated computation related to anchor boxes such as the IOU computation and matching between the anchor boxes and ground-truth boxes during training. As a results, FCOS shows faster inference speed and less training memory footprint than RefineDet. The pure training sample selection scheme of ATSS results in the smallest number of parameters with the fastest inference speed. This is similar to the finding in [36]. The computational complexity of ATSS lies in the use of tiling multiple anchors per location on the image to detect objects. The computational complexity of our method comes from the regressor head, classifier head and backbone. Because although sharing a common structure, the classifier head and the regressor head use separate parameters. Following ATSS, the computational complexity of our proposed PSTD-Net remains at the second lowest levels. Table 4 summarizes the result of the computational complexity comparison.

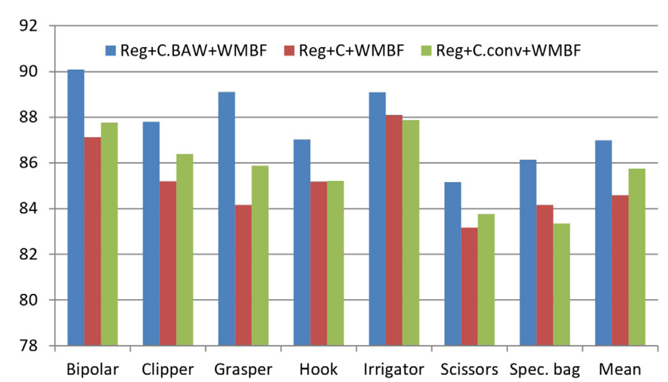

Fig. 8. Effect of Bi-directional Adaption Weighting (Reg+C.BAW+WMBF) compared with convolutional layers (Reg+C.conv+WMBF) and traditional classifier (Reg+C+WMBF).

4.5. Ablation study

We carry out ablation experiments to better understand the effect of specific components of PSTD-Net.

4.5.1. Bi-directional Adaption Weighting (BAW)

To investigate the role of BAW, we design a comparison method called Reg+C.Conv+WMBF, which replaces the BAW module with 2 convolutional layers. Another comparison method

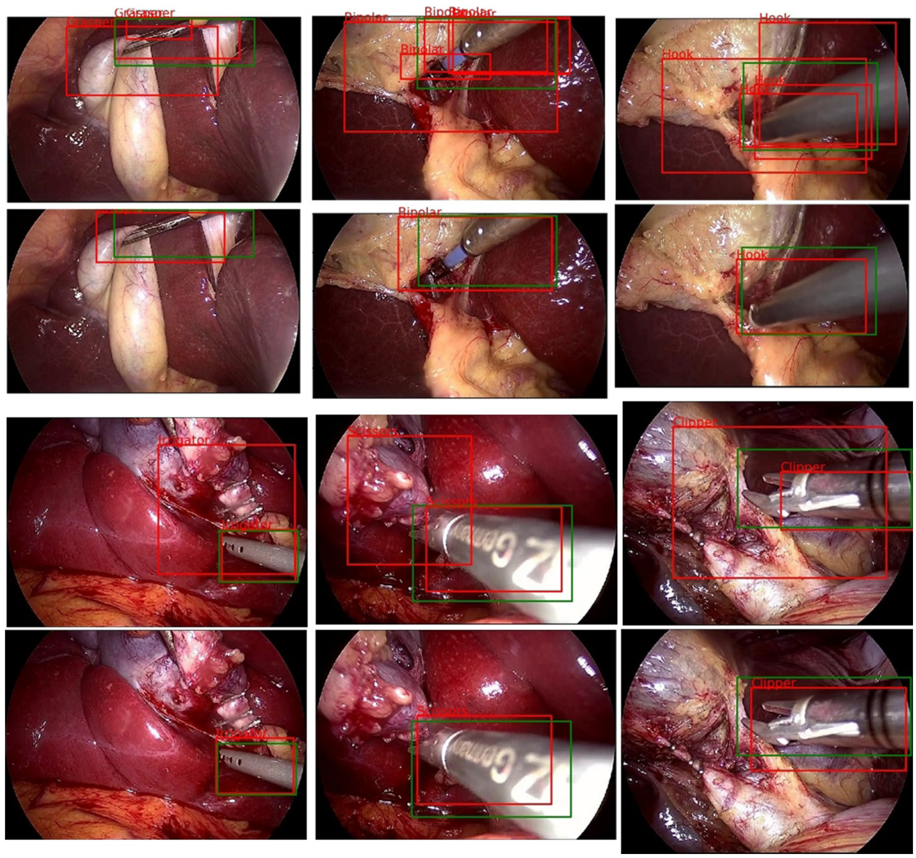

Fig. 9. Detection results before (left) and after (right) WMBF. Red box: detection, green box: ground-truth.

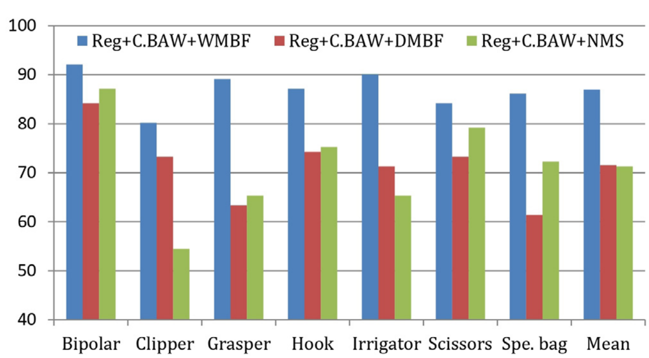

Fig. 10. Effect of Bounding Boxes Fusion strategies, AP for 7 classes and mean AP of 3 different Box Fusion strategies.

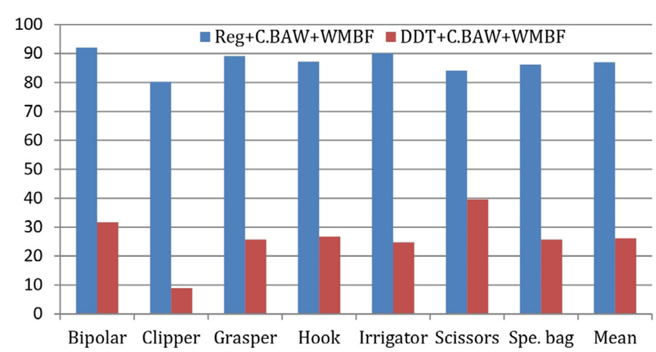

Fig. 11. Effect of Bounding Boxes Regressor, AP for 7 classes and mean AP.

is Reg+C+WMBF, which directly localizes tools by convolution layers + bounding box regression head without the use of BAW module in the classifier. The proposed entire model (Reg+C.BAW+WMBF) obtains mAP at 86.99. While the mAP values of the two comparison methods (Reg+C+WMBF and Reg+C.Conv+WMBF) are 84.58 and 85.74 respectively (see Fig. 8).

Reg+C.Conv+WMBF only obtains 1% mAP improvement over Reg+C+WMBF. This results from the addition of the convolutional layers that promote the feature extraction and fusion in single channel. However, The PSTD-Net still presents obviously higher mAP compared with Reg+C.Conv+WMBF. This demonstrates the significance of emphasizing informative channels that have proper activation to the target tools, which is exact the motivation of the BAW module. As we know, without channel-wise weighting, the contributions of distinct channels are equal. In fact, due to different activation properties, channels shows quite distinct responses to tools and background tissues.

4.5.2. Bounding Boxes Fusion strategies

In order to figure out the contribution of our Weighted Mean Box Fusion (WMBF) strategy, we carry out a comparison experiment between WMBF, Direct Mean Box Fusion (DMBF) and Non Maximum Suppress (NMS) respectively. DMBF is directly using the center of the top-left and bottom-right corner of the bounding boxes as the top-left and bottom-right corner of the output bounding box of a specific tool. NMS is a traditional strategy for removing the redundancy of bounding boxes. We set the IoU threshold in NMS as 0.1.

Fig. 9 illustrates tool detection results before (left column) and after (right column) WMBF. To better understand the comparison,

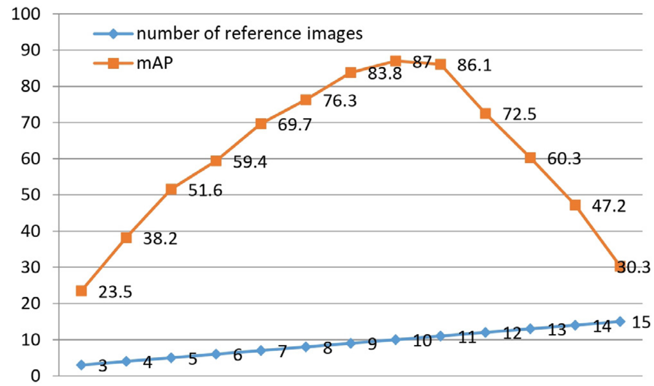

Fig. 12. The performance of PSTD when the number of reference images increases from 3 to 15.

please see the bottom pipeline in Fig. 1 as a reference. The left column shows the boxes predicted by the regressor, where multiple overlapping boxes may exist and inaccurate boxes may also occur. WMBF fuses the coordinates of the multiple input boxes with the consideration to boxes' confidence scores. The right column visualizes the processed boxes, which are spatially closer to the ground-truth in terms of sizes and centroid positions.

As shown in Fig. 10, NMS gets the lowest mAP (71.29%) among all three strategies. Due to the noise from DDT, the confident scores given by the regressor are not clear enough for direct use by NMS, which highly depends on pure confident scores. This is the main reason why NMS performs poorly here. As NMS only keeps the bounding box with the highest score within redundant area, the precision of NMS will totally depend on the unreliable score obtained during bounding box generation. Thus the reliability of NMS is greatly affected in a noisy system.

DMBF gets a slightly higher mAP (71.57%) than NMS. This gap attributes to the contribution of the bounding boxes that get lower confident scores but locate quite close to the ground truth. In these cases, DMBF breaks through NMS's drawback of over-reliance on the reliability of boxes' confident scores. On the other hand, averaging is able to smooth the output noise to some extent. However, DMBF absolutely neglects the information of confident scores from the regressor. As Regressor is actually a bandpass filter, it can somehow suppress the noise in the training samples generated by DDT. The confident score still contains useful messages in a large scale. It is the ignorance of the information in bounding box confident score that results in DMBF's insignificant advantage over NMS even though it reduces the disturbance of noise.

Unlike NMS and DMBF, WMBF can not only smooth the noise but also utilize the useful information from the regressor. As a result, WMBF achieves 86.99% mAP with ∼16% higher than the other two strategies.

4.5.3. Bounding Boxes Regressor

In PSTD-Net, we train a bounding box regressor to perform real tool detection, and to further refine the pseudo bounding boxes generated by DDT. In this section, we construct a no-regressor pipeline by removing the regressor module and concatenating DDT with the classifier directly. To be fair, we still use the green-background reference set mentioned in Section 3.2 for this no-regressor pipeline. Fig. 11 shows the performance of Reg+Cbaw+WMBF (our complete pipeline) and DDT+Cbaw+WMBF (no-regressor pipeline). We can observe a huge mAP gap (close to 60%) between the two pipelines. The main reason for this phenomenon is that the bounding boxes generated by DDT contain a lot of noise in terms of boxes locations and sizes. In the complete pipeline, the bounding box regressor is designed to refine the bounding boxes from DDT. Without the regressor, noises in boxes locations and sizes can directly pass to the subsequent classifier, causing the significant mAP drop in DDT+C.BAW+WMBF.

4.5.4. Number of reference images

During pseudo bounding box generation, the input follows a $1+N$ mode (1 input image and N reference images belonging to the same category with the input image). The motivation of the $1+N$ mode is to purify the noisy bounding boxes among the pseudo bounding boxes generated solely by DDT. Green background reference images are used to suppress interferences from original background. Within the $1+N$ mode, input image areas that are similar to or simultaneously shared by all the N reference images will be selected as region proposal (i.e. the generated pseudo bounding boxes). This is more reliable than principal component analysis done by pure DDT.

If we set the value of N properly, every training image can get higher-quality tool bounding boxes, which focus more on the regions where surgical tools are present. If we set N too small, it means the constrain is loose. Thus, many shared but irrelevant image areas will be given as pseudo bounding boxes. If we set N too large, the constrain will be strict. Thus, bland image area will be given as pseudo bounding boxes.

Here we conduct a new experiment to explore how N affects the proposed method. We set N from 3 to 15, and report the mAP performance under every value of N. Fig. 12 shows this experiment result. It can be seen the trend of mAP presents three close and high values when $N = 9, 10, 11$. This is consistent with our previous theoretical analysis.

5. Conclusion

In this paper, we propose a pseudo supervised surgical tool detection (PSTD) framework to solve the drawbacks in existing surgical tool detection methods. Furthermore, our Bi-directional Adaption Weighting (BAW) mechanism brings further improvement on the basis of PSTD by deep mining into the convolution features. Various experiments show that our methods obtain a significant edge over previous methods, including the recent state-of-the-art method EndoNet [5]. By three level of explicit localization refinement measures (pseudo bounding box generation, real box regression, weighted boxes fusion) and the tool classifier with BAW mechanism, we witness the effectiveness to use image level tool category labels for tool detection, without the need to relying on pixel level tool annotations which are rare and expensive. We believe this finding is encouraging for future relevant research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] T.T. Kim, J.P. Johnson, R. Pashman, D. Drazin, Minimally invasive spinal surgery with intraoperative image-guided navigation, BioMed. Res. Int. 2016 (2016) 1–7.
[2] Z. Zhou, M.M.R. Siddiquee, N. Tajbakhsh, J. Liang, Unet++: Redesigning skip connections to exploit multiscale features in image segmentation, IEEE Trans. Med. Imaging (2019).
[3] S. Moccia, S. Foti, A. Routray, F. Prudente, A. Perin, R. Sekula, L. Mattos, J. Balzer, W. Fellows, E. De Momi, C. Riviere, Toward improving safety in neurosurgery with an active handheld instrument, Ann. Biomed. Eng. 46 (2018) http://dx.doi.org/10.1007/s10439-018-2091-x.

[4] D. Sarikaya, J.J. Corso, K.A. Guru, Detection and localization of robotic tools in robot-assisted surgery videos using deep neural networks for region proposal and detection, IEEE Trans. Med. Imaging 36 (2017) 1542–1549.

[5] A.P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, N. Padoy, EndoNet: A deep architecture for recognition tasks on laparoscopic videos, IEEE Trans. Med. Imaging 36 (1) (2017) 86–97.

[6] L. Yu, P. Wang, X. Yu, Y. Yan, Y. Xia, A holistically-nested U-net: Surgical instrument segmentation based on convolutional neural network, J. Digital Imaging 33 (2) (2020) 341–347.

[7] F. Despinoy, D. Bouget, G. Forestier, C. Penet, N. Zemiti, P. Poignet, P. Jannin, Unsupervised trajectory segmentation for surgical gesture recognition in robotic training, IEEE Trans. Biomed. Eng. 63 (6) (2016) 1280–1291.

[8] Y. Xue, G. Bigras, J. Hugh, N. Ray, Training convolutional neural networks and compressed sensing end-to-end for microscopy cell detection, IEEE Trans. Med. Imaging 38 (11) (2019) 2632–2641, http://dx.doi.org/10.1109/TMI.2019.2907093.

[9] N. Rieke, D.J. Tan, M. Alsheakhali, F. Tombari, C. Filippo, V. Belagiannis, A. Eslami, N. Navab, Surgical tool tracking and pose estimation in retinal microsurgery, in: Medical Image Comput. Comput. Assist. Intervention, vol. 9349, MICCAI, 2015, pp. 266–273, http://dx.doi.org/10.1007/978-3-319-24553-9_33.

[10] N. Rieke, D.J. Tan, C.A. di San Filippo, F. Tombari, M. Alsheakhali, V. Belagiannis, A. Eslami, N. Navab, Real-time localization of articulated surgical instruments in retinal microsurgery, Med. Image Anal. 34 (2016) 82–100, http://dx.doi.org/10.1016/j.media.2016.05.003.

[11] X. Du, M. Allan, A. Dore, S. Ourselin, D. Hawkes, J.D. Kelly, D. Stoyanov, Combined 2D and 3D tracking of surgical instruments for minimally invasive and robotic-assisted surgery., Int. J. Comput. Assist. Radiol. Surg. 11 (6) (2016) 1109–1119.

[12] M. Allan, S. Ourselin, S. Thompson, D.J. Hawkes, J. Kelly, D. Stoyanov, Toward detection and localization of instruments in minimally invasive surgery, IEEE Trans. Biomed. Eng. 60 (4) (2013) 1050–1058.

[13] A. Reiter, P. Allen, T. Zhao, Appearance learning for 3D tracking of robotic surgical tools, Int. J. Robot. Res. 33 (2014) 342–356, http://dx.doi.org/10.1177/0278364913507796.

[14] R. Stauder, A. Okur, L. Peter, A. Schneider, M. Kranzfelder, H. Feussner, N. Navab, Random forests for phase detection in surgical workflow analysis, in: Information Processing in Computer-Assisted Interventions, IPCAI, 2014, pp. 148–157, http://dx.doi.org/10.1007/978-3-319-07521-1_16.

[15] R. Sznitman, C. Becker, P. Fua, Fast part-based classification for instrument detection in minimally invasive surgery, in: Medical Image Comput. Comput. Assist. Intervention, MICCAI, 2014, p. 692.

[16] L. Garcia-Peraza-Herrera, W. Li, L. Fidon, C. Gruijthuijsen, A. Devreker, G. Attilakos, J. Deprest, E. Vander Poorten, D. Stoyanov, T. Vercauteren, S. Ourselin, ToolNet: Holistically-nested real-time segmentation of robotic surgical tools, RSJ Int. Conf. Intell. Robots Syst.(IROS) (2017) 5717–5722.

[17] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B.B. Haro, L. Zappella, S. Khudanpur, R. Vidal, G.D. Hager, A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery, IEEE Trans. Biomed. Eng. 64 (9) (2017) 2025–2041.

[18] X. Zhang, Y. Yang, Y. Wei, T. Huang, J. Feng, Adversarial complementary learning for weakly supervised object localization, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 1325–1334.

[19] Y. Zhang, Y. Bai, M. Ding, Y. Li, B. Ghanem, Weakly-supervised object detection via mining pseudo ground truth bounding-boxes, Pattern Recognit. 84 (2018) 68–81.

[20] J. Choe, H. Shim, Attention-based dropout layer for weakly supervised object localization, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2019, 2019, pp. 2214–2223.

[21] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Computer Vision and Pattern Recognition (CVPR), in: 2016 IEEE Conference on, 2016, pp. 2921–2929.

[22] K.K. Singh, Y.J. Lee, Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization, in: International Conference on Computer Vision, ICCV, 2017.

[23] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. PP (99) (2017).

[24] H. Wang, Z. Ji, Z. Lin, Y. Pang, X. Li, Stacked squeeze-and-excitation recurrent residual network for visual-semantic matching, Pattern Recognit. 105 (2020).

[25] Z. Fan, G. Chen, J. Wang, H. Liao, Spatial position measurement system for surgical navigation using 3-D image marker-based tracking tools with compact volume, IEEE Trans. Biomed. Eng. 65 (2) (2018) 378–389.

[26] D. Bouget, R. Benenson, M. Omran, L. Riffaud, B. Schiele, P. Jannin, Detecting surgical tools by modelling local appearance and global shape, IEEE Trans. Med. Imaging 34 (12) (2015) 2603–2617.

[27] E. Colleoni, S. Moccia, X. Du, E. De Momi, D. Stoyanov, Deep learning based robotic tool detection and articulation estimation with spatio-temporal layers, IEEE Robot. Autom. Lett. 4 (3) (2019) 2714–2721.

[28] D. Bouget, M. Allan, D. Stoyanov, P. Jannin, Vision-based and marker-less surgical tool detection and tracking: a review of the literature, Med. Image Anal. 35 (2017) 633–654.

[29] S. Wu, Y. Xu, DSN: A new deformable subnetwork for object detection, IEEE Trans. Circuits Syst. Video Technol. PP (2019) 1, http://dx.doi.org/10.1109/TCSVT.2019.2905373.

[30] N. Rieke, D.J. Tan, C. Amat di San Filippo, F. Tombari, M. Alsheakhali, V. Belagiannis, A. Eslami, N. Navab, Real-time localization of articulated surgical instruments in retinal microsurgery, Med. Image Anal. 34 (2016) 82–100.

[31] Y. Xue, Y. Li, S. Liu, X. Zhang, X. Qian*, Crowd scene analysis encounters high density and scale variation, IEEE Trans. Image Process. 30 (2021) 2745–2757, http://dx.doi.org/10.1109/TIP.2021.3049963.

[32] Y. Wang, Y. Wei, X. Qian, L. Zhu, Y. Yang, AINet: Association implantation for superpixel segmentation, in: International Conference on Computer Vision, ICCV, 2021.

[33] X. Li, S. Lai, X. Qian*, DBCFace: TOwards pure convolution neural network face detection, IEEE Trans. Circuits Syst. Video Technol. (2021) http://dx.doi.org/10.1109/TCSVT.2021.3082635.

[34] Z. Tian, C. Shen, H. Chen, T. He, FCOS: Fully convolutional one-stage object detection, in: The IEEE International Conference on Computer Vision, ICCV, 2019.

[35] S. Zhang, L. Wen, X. Bian, Z. Lei, S.Z. Li, Single-shot refinement neural network for object detection, IEEE Trans. Circuits Syst. Video Technol. 31 (2) (2021) 674–687.

[36] S. Zhang, C. Chi, Y. Yao, Z. Lei, S.Z. Li, Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020.

[37] H. Bilen, A. Vedaldi, Weakly supervised deep detection networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[38] P. Tang, X. Wang, X. Bai, W. Liu, Multiple instance detection network with online instance classifier refinement, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 3059–3067.

[39] M. Gao, A. Li, R. Yu, V.I. Morariu, L.S. Davis, C-WSL: Count-guided weakly supervised localization, in: The European Conference on Computer Vision, ECCV, 2018.

[40] G. Cheng, J. Yang, D. Gao, L. Guo, J. Han, High-quality proposals for weakly supervised object detection, IEEE Trans. Image Process. 29 (2020) 5794–5804.

[41] C.L. Zhang, Y.H. Cao, J. Wu, Rethinking the route towards weakly supervised object localization, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2020.

[42] S. Xiu, C. Zhang, S.C. Wu Jianxin, Z. Zhihua, Unsupervised object discovery and co-localization by deep descriptor transformation, Pattern Recognit. 88 (2019) 113–126.

[43] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778.

[44] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, 2021, arXiv:2103.14030.

[45] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Trans. Pattern Anal. Mach. Intell. PP (99) (2017) 2999–3007.

[46] S. Ren, K. He, R. Girshick, J. Sun, Faster r-CNN: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems 28, 2015, pp. 91–99.

[47] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.