Surgical tool detection via multiple convolutional neural networks

Huoling Luo, Qingmao Hu, Fucang Jia

Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Email: fc.jia@siat.ac.cn

1 Introduction

Surgical tool detection and analysis are important to surgical workflow recognition and video indexing. Recently, convolutional neural network (CNN) has gained a huge success in computer vision applications, especially in object detection [1] and image classification [2]. CNN is a powerful tool that can retrieve hierarchical vision of features. Based on the M2CAI 2016 tool challenge requirements and its released training data, a CNN based method is proposed which fully leverages the training data to obtain a better surgical tool detection accuracy. This method is inspired from a newly published paper [3], which introduced CNN (EndoNet) for laparoscopic videos to carry out the surgical phase recognition and tool presence detection. The Keras software [4] was used to implement the proposed method.

2 CNN Architecture

The proposed method consists of seven CNNs; each represents one tool to be detected as illustrated in Fig. 1. The architecture of CNN for each tool is the same shown in details in Fig. 2, which consists of an input layer, three convolutional layers (C1 to C3), and two fully-connected layers (F4 and F5). The output is connected to a binary classifier with output either 0 or 1.

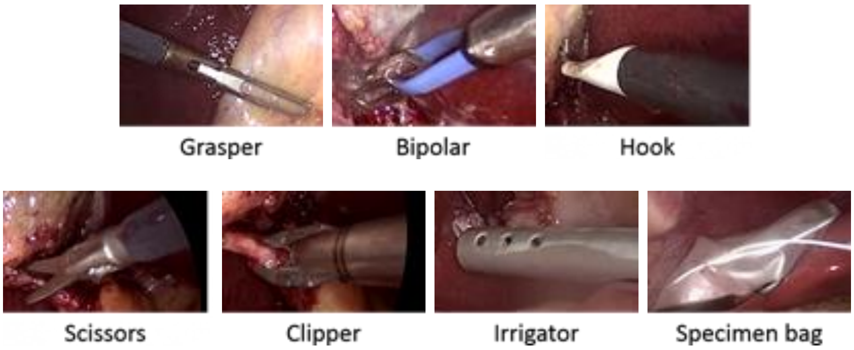

Fig. 1. Seven different types of tools required to detect (adopted from [3])

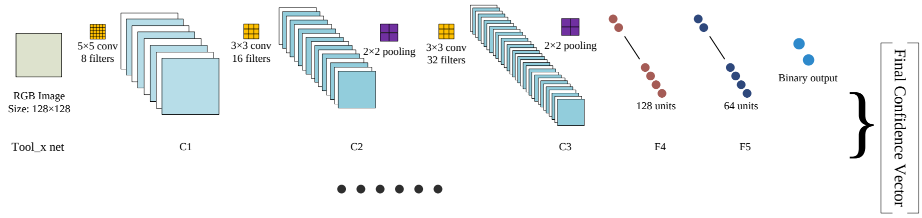

Fig. 2. Illustration of the proposed surgical tool detection CNNs architecture. Tool_x denotes Grasper, Bipolar, Hook, Scissors, Clipper, Irrigator and Specimen bag, respectively. The input of the CNNs is fixed 128x128 using a ReLU activation function. The layer C1 consists of eight feature maps with convolutional filter 5x5 while layer C2 has 16 feature maps, and its filter size is 3x3 with the following pooling layer that has size 2x2. Layer C3 includes 32 feature maps with filter size 3x3 and followed by pooling layer with size 2x2. Layers F4 and F5 are fully connected, and the output of F5 is either 0 or 1 to verify the tool presence in the input image.

3 Training Stage

We divide the training data into seven groups; every group includes positive and negative samples according to the ground truth given in the text files (See Table 1). It is clearly showed that, the distribution of positive and negative samples is imbalanced except grasper and hook tool types. In order to process the imbalance of training samples, we re-extract the frames based on the ground truth text files from training videos and construct the final training data shown in Table 2.

Table 1. The training data distributions of different types of tool networks

	Positive Samples	Negative Samples
Grasper	10967	12320
Bipolar	635	22651
Hook	14130	9156
Scissors	411	22875
Clipper	878	22408
Irrigator	954	22333
Specimen Bag	1498	21782

Table 2. The training data distributions re-extracted from original training videos

	Positive Samples	Negative Samples
Grasper	10967	11200
Bipolar	11335	11325
Hook	10597	9156

Scissors	9748	10166
Clipper	11677	11204
Irrigator	11269	11166
Specimen Bag	10168	10891

4 Testing Stage

For a given test frame, we first resize it to fit the input of every CNNs and then test it by the seven nets sequentially. The final detection result is consisted of the seven different types of CNN.

References

1. R. Girshick, J. Donahue, T. Darrell, J. Malik. Region-based convolutional networks for accurate object detection and segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016. 38(1), 142-158.
2. A. Krizhevsky, I. Sutskever, G.E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems. 2012, 1097-1105.
3. 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 Transactions on Medical Imaging. 2016. doi: 10.1109/TMI.2016.2593957
4. F. Chollet. Keras. GitHub repository: https://github.com/fchollet/keras. 2015.
5. M2CAI Challenge with MICCAI 2016. http://camma.u-strasbg.fr/m2cai2016/index.php/program-challenge/