Learning Latent Temporal Connectionism of Deep Residual Visual Abstractions for Identifying Surgical Tools in Laparoscopy Procedures

Kaustuv Mishra, Rachana Sathish and Debdoot Sheet
Department of Electrical Engineering
Indian Institute of Technology Kharagpur, West Bengal, India
{kaustuvmishra, rachanasathish}@iitkgp.ac.in, debdoot@ee.iitkgp.ernet.in

Abstract

Surgical workflow in minimally invasive interventions like laparoscopy can be modeled with the aid of tool usage information. The video stream available during surgery primarily for viewing the surgical site using an endoscope can be leveraged for this purpose without the need for additional sensors or instruments. We propose a method which learns to detect the tool presence in laparoscopy videos by leveraging the temporal connectionist information in a systematically executed surgical procedures by learning the long and short order relationships between higher abstractions of the spatial visual features extracted from the surgical video. We propose a framework consisting of using Convolutional Neural Networks for extracting the visual features and Long Short-Term Memory network to encode the temporal information. The proposed framework has been experimentally verified using a publicly available dataset consisting of 10 training and 5 testing annotated videos to obtain an average accuracy of 88.75% *in detecting the tools present.*

1. Introduction

Laparoscopy or key hole surgical procedures are practiced widely in the clinics owing to lower patient discomfort and a faster post-surgical recovery time. Lack of direct access to the surgical site, indirect mode of visualizing the site using an endoscope, restricted freedom of movement of the tools and the very nature of the tools used, makes the procedure challenging. Developing context-aware smart systems to aid the surgeon requires information regarding the tool usage along with other pathological and anatomical information. This paper presents a framework for automatic detection of tools being used at each instance of a surgery using the video stream recorded during the surgery. The proposed framework does not require markers or sensors for detection of tool usage. Tool usage information can be further used for modeling the surgical workflow and evaluation of surgical skill. The overview of the proposed framework is presented in Fig. 1

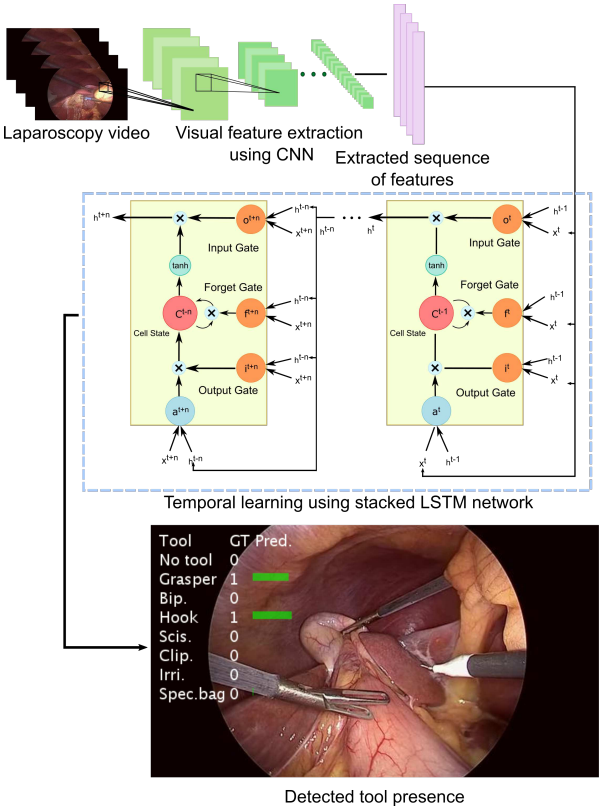


Figure 1. Overview of the proposed method.

Challenges: In laparoscopy video, the tools often appear occluded which renders the task of identifying them challenging. Another major challenge is the mobile nature of the imaging device which contributes to motion artifacts. Additionally, the dynamically changing nature of illumination used during surgery and the lack of background texture due to smooth tissue surfaces also causes specular reflections which are comparable to the appearance of the metal-

(a) Grasper (b) Bipolar (c) Hook (d) Clipper (e) Scissors (f) Irrigator (g) Specimen Bag

Figure 2. The seven types of tools used for performing various surgical tasks in the video viz., (a) Grasper, (b) Bipolar, (c) Hook, (d) Clipper, (e) Scissors, (f) Irrigator and (g) Specimen bag.

lic surface of the tools used.

Approch: The appearance of surgical tools are quite different from the visual appearance of internal structures of the human body. Visual features which captures these differences can be leveraged for the detection of tools. In the proposed framework, a Convolutional Neural Network (CNN) [9] which is capable of learning the high level visual representation in images is used to extract visual features from the frames of the surgical video. Since the spatial features extracted using the CNN do not incorporate the temporal information, they fail to learn connectionism across neighboring frames. The various procedures in a surgery are almost always executed methodically and use an order of tools depending on the expertise of the surgeon and the complications that may arise during the surgery. Thus, the temporal information being vital for accurate detection of tool presence, we also train a long short-term memory (LSTM) [5] on the extracted spatial features of the video sequence to capture the temporal connectionism across deep residual visual features and thereby increase the accuracy in prediction.

This paper is organized as follows. The existing techniques for detection of surgical tool usage is briefly described in Sec. 2. The challenge at hand is formally defined in Sec. 3 . The methodology is explained in Sec. 4 . The experiments are detailed with the results in Sec. 5. Sec. 6 discusses the results obtained. The conclusion is presented in Sec. 7

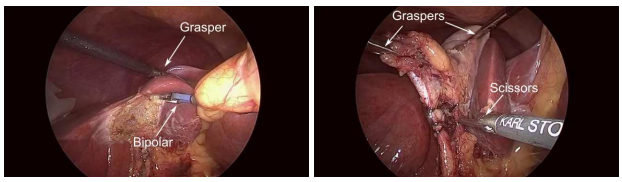
2. Prior Art

Some of the initial efforts on detection and tracking of tools in minimally invasive surgery focused on physically tagging the tools with additional sensors to track their usage. Typically, pattern tags [2], color tags [16, 13], LED [8] and RFID tags [10] are used as markers. These methods require physical modification of the existing operating instrument and set up, which involves integration of markers to the tools thereby making them bulkier. Also installation of additional sensors to detect these markers tends to hamper degree of flexibility required by the surgeon. Image processing based techniques for tool detection which operate on the video recorded using the endoscope during the surgery eliminates the need for additional markers or sensors. Sznitman *et al*. [15] had proposed an instrument-part detector based on gradient boosted regression trees for detecting tools in minimally invasive surgery. Template matching based localization and estimation of pose of surgical tools was proposed by Reiter *et al*. [11]. Image processing techniques like k-means clustering [12] and Kalman filtering have also been used for localization and tracking of tools in surgical video. Twinanda *et al*. [17] have used convolutional neural network and hidden markov model (HMM) for detecting tools in laparoscopy videos.

These methods either either require integration of additional hardware to the existing operating room set up which reduces the degree of flexibility, or requires initialization of the tool tracker in the initial frame of the video. Also, these methods does not use temporal similarity across frames on a local scale for detection of tool presence.

3. Problem Statement

The surgical procedures involved in a laparoscopy surgery are performed using multiple tools. Under a typical scenario, they consist of grasper, bipolar, hook, clipper, scissors, irrigator and specimen bag which are shown in Fig. 2. Thus, each frame in the video recording of such surgeries may contain multiple tools up to three such as shown in Fig. 3, whose presence is to be detected. The problem of multiple tool presence detection in the video of certain laparoscopy surgery performed using a set of tools $T = \{t_1, t_2, \ldots, t_n\}$ can be therefore formally defined as a multi-label multi-class classification task where in each frame f in the video we have to detect the set of tools T_f which are present in the frame out of the total set of tools T being used in the procedure.



(a) Two tools (Grasper and bipolar) (b) Three tools (Two graspers and scissors)

Figure 3. Sample frames from the dataset showing simultaneous presence of (a) two tools and (b) three tools.

4. Exposition to the Solution

We propose a solution using CNN and LSTM which utilizes both spatial visual abstractions and their temporal connectionism in surgical videos. CNNs have been successfully used in image recognition tasks and are used here for extracting spatial visual features to aid the multi-label multi-class tool presence detection task by means of transfer learning. The features extracted by the CNN are then used to train an LSTM to accurately detect the tool presence by incorporating the temporal dimension of information.

4.1. CNN for multiple tool detection

Training a deep CNN architecture from start for detecting tools in surgical video is challenged by the limited availability of annotated surgical data and also by limited visual variations. Therefore, we use CNN architectures pre-trained on the large dataset like ImageNet for Large Scale Visual Recognition Challenge (ILSVRC) [3]. The classifier layer in the pre-trained CNNs is replaced and the entire network is then fine-tuned on the task specific dataset with fewer annotated data to learn visual features that differentiates the appearance of the surgical tools from the background. We have used ResNet-50 [4] in our proposed method. The framework for extracting visual attributes used is shown in Fig. 4. Frames of size 224×224px are provided as input to CNN and the features from the last convolutional layer of ResNet-50 is flattened to obtain a feature of length $2,048$. The network is then augmented with a fully-connected layer of length 8 and a log-sigmoid classification layer with transfer function defined as,

$$f(x) = \log\left\{\frac{1}{1+\exp(-x)}\right\} \tag{1}$$

where x is the input to the layer. Unlike the soft-max classifier which is typically used for the single-label classification task, the log-sigmoid layer allows assigning of multiple labels to an image.

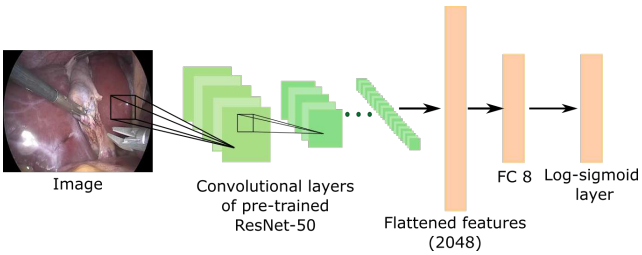


Figure 4. Architecture of the network used.

The CNN is trained using a multi-label multi-class loss based on binary cross-entropy defined accordingly,

$$L(\mathbf{x},\mathbf{y}) = \frac{-1}{N}\sum_{i=1}^{N}\left[\mathbf{y}(i)\log\left\{\frac{\exp(\mathbf{x}(i))}{1+\exp(\mathbf{x}(i))}\right\} + (1-\mathbf{y}(i))\log\left\{\frac{1}{1+\exp(\mathbf{x}(i))}\right\}\right] \tag{2}$$

where, $\mathbf{x}(i)$ is the prediction for each class, $\mathbf{y}$ is the binary ground-truth annotation for tool presence with $\mathbf{y}(i) \in \{0,1\}$ and N is the number of classes.

4.2. LSTM for tool detection in video sequence

The features learned by last layer of the CNN before the classification layer is used to train a deep LSTM network. In order to capture the temporal redundancies across the spatial visual features across neighboring frames fed sequentially to the network we stack three LSTM blocks. A soft-sign layer is added at the end of the network as shown in Fig.5. Each LSTM network is a vanilla-LSTM [5] comprising of three gates viz., input gate, forget gate and output gate and no peephole connections. The output of the input gate $\mathbf{i}_t$, forget gate $\mathbf{f}_t$, cell state $\mathbf{C}_t$, output gate $\mathbf{o}_t$ and hidden state $\mathbf{h}_t$ at instance t is given as,

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{3}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{4}$$

$$\widetilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C\mathbf{x}_t) + \mathbf{W}_C\mathbf{h}_{t-1} + \mathbf{b}_C \tag{5}$$

$$\mathbf{C}_t = \mathbf{f}_t\mathbf{C}_{t-1} + \mathbf{i}_t\widetilde{\mathbf{C}}_t \tag{6}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \tag{8}$$

where, $\sigma(\cdot)$ is the sigmoid function, $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o, \mathbf{W}_C$ are the weights corresponding to the gates and the states, $\mathbf{h}_{t-1}$ is the hidden state at previous time instance, $\mathbf{x}_t$ is the input at current instance, $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o, \mathbf{b}_C$ are the biases corresponding to the gates and the states and $\widetilde{\mathbf{C}}_t$ is the new cell state value that can be added to $\mathbf{C}_t$.

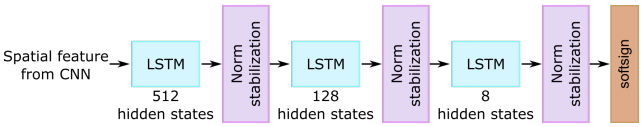


Figure 5. Architecture of the stacked LSTM network used for tool detection in surgical videos.

The input to the deep LSTM network is a sequence of visual features from 10 sequential frames from the video. These visual features are extracted by the network detailed in Sec. 4.1. The number of hidden states in each of the LSTM modules are 512, 128 and 8, where the

number of states is progressively reduced with depth. As shown in Fig. 5, each LSTM network is followed by norm-stabilization [7]. It regularizes the hidden states of the LSTM by minimizing the difference between the L2 norm of hidden states at consecutive steps. The cost function used here is defined as,

$$L = \beta \frac{1}{T} \sum_{t=1}^{T} (\|\mathbf{h}_t\|_2 - \|\mathbf{h}_{t-1}\|_2)^2 \tag{9}$$

where, $\mathbf{h}_t$ is the hidden state at time step t and β is the hyper-parameter controlling the amount of regularization. In the last layer of the deep network, $softsign$ [1] transfer defined as below is applied on the output of the LSTM.

$$softsign(x) = \frac{x}{1 + |x|} \tag{10}$$

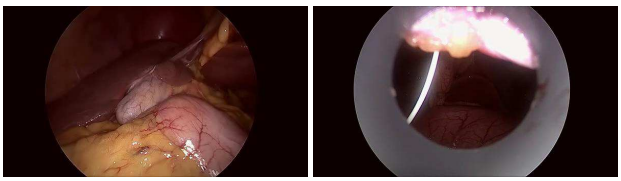
where, x is the input. The network is trained using the multi-label multi-class loss described in (2).

5. Experiments and Results

5.1. Dataset description

The proposed method is experimentally verified on the m2ccai16-tool[1] dataset. The dataset comprises of 15 laparoscopy videos (including 10 training and 5 testing sets) of cholecystectomy procedures and the tool class binary annotation for tool presence in every 25^{th} of the videos recorded at 25fps. Fig. 2 shows the seven types of tools that were used in these videos while performing various surgical tasks.

In the videos provided in the dataset, it was observed that during the initial stage of the procedure when the surgeon is preparing for the surgery and the final stage when the surgeon is withdrawing the endoscope from the incision, no tools appear in the frame as shown in Fig. 7(a) and Fig. 7(b) respectively. Therefore, we consider an additional class in the multi-label multi-class classification problem which corresponds to no tool being present in the frame so as to decrease the false positive detections.



(a) Start of surgery with no tool (b) End of the surgery with no tool

Figure 7. Sample frames from a video without any tool present.

[1] http://camma.u-strasbg.fr/datasets

5.2. Compensating class imbalance

The various surgical tasks in the cholecystectomy procedure are characterized by the usage of certain set of tools. Due to varying duration of these tasks, certain tools occur more frequently across all the videos resulting in severe class imbalance in the dataset as shown in Fig. 6(a). The figure graphically illustrates the proportion of each tool and their combination present in the dataset. Sections connected to a single tool indicates the number of frames in which the tool is present by itself and strips joining two different tools indicates the number of frames in which they occur together. This includes frames with two as well as three tools.

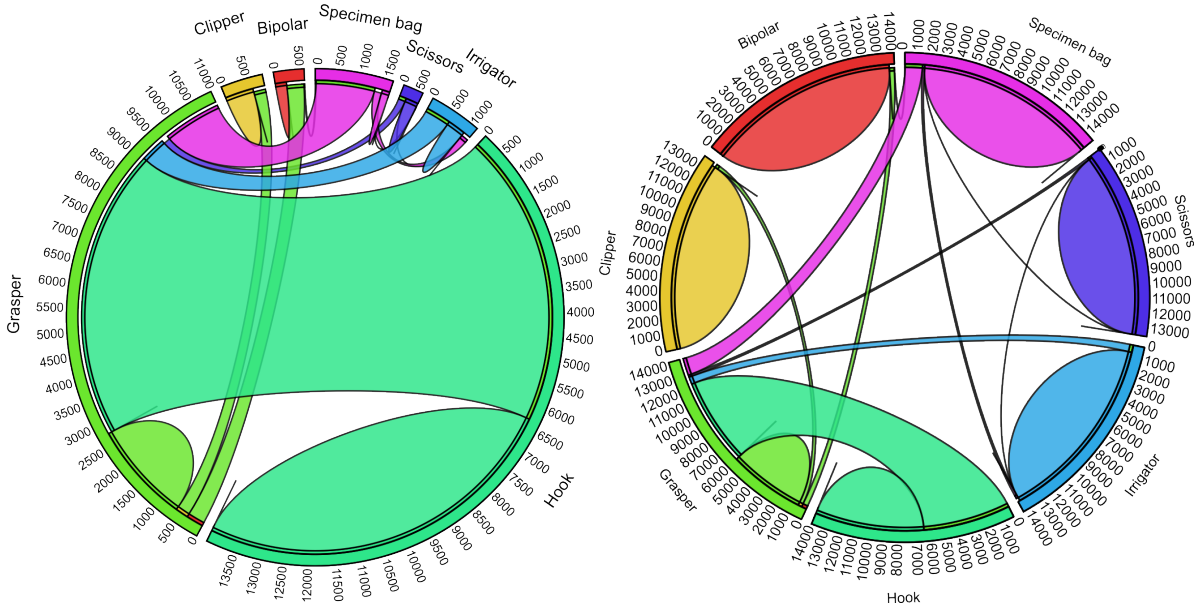
In order to compensate for this imbalance we methodically extract frames from the videos such that the number of frames containing each of the tools are approximately same. Since the annotation in the dataset is provided at 1fps while the actual video is recorded at 25fps, the class imbalance can be compensated by considering the frames which are not annotated and interpolating the existing annotation to these frames. As an example, there are 411 annotated frames in the training set that contain scissors. We balance the data for this class by extracting the intermediate frames between the annotated frames. Data augmentation by flipping the frames from left to right, top to bottom and their combination is also done when required to balance the data for a class. Exactly equal proportion could not be achieved due to co-occurance of various tools. Tab. 2 shows the number of annotated frames containing each of the tool in the raw training data and the class-balanced training data. The proportion of occurrence of various tools and their combinations in the balanced data is graphically illustrated in Fig. 6(b).

Tool	Raw data	Balanced data
No tool	2,749	14,321
Grasper	10,967	14,342
Bipolar	635	14,327
Hook	14,130	14,110
Scissors	411	13,449
Clipper	878	13,928
Irrigator	953	14,231
Specimen bag	1,504	14,495

Table 2. Comparison of number of frames containing each class of tool in the raw training data and balanced training data.

5.3. Training parameters

The CNN detailed in Sec. 4.1 is trained on the class balanced data with a learning rate 1×10^{-4}, weight decay parameter 1×10^{-4}, momentum 0.9 and batch size 64. The network is optimized using stochastic gradient descent algorithm (SGD).



(a) Raw data (b) Balanced training data

Figure 6. Proportion of occurrence of the different tools in (a) raw and (b) balanced training data.

Baseline	Description	Train. time per epoch (min)	Test. time per frame (ms)
BL1	Modified (multi-label multi-class) AlexNet[6]	18.00	0.40
BL2	Modified (multi-label multi-class) AlexNet[6] (BL1) + LSTM	18.33	1.34
BL3	Modified (multi-label multi-class) GoogLeNet[14]	23.00	0.94
BL4	Modified (multi-label multi-class) GoogLeNet[14] (BL3) + LSTM	23.17	1.20
BL5	Modified (multi-label multi-class) ResNet-50[4]	35.00	1.95
Proposed Method	Modified (multi-label multi-class) ResNet-50[4] (BL5) + LSTM	35.30	2.42

Table 1. Baselines for performance comparison.

5.4. Baselines

To evaluate the performance of the proposed method, we have considered six baselines (BL) for comparison as summarized in Tab. 1.

5.5. Implementation

The proposed method was implemented and evaluated using Torch[2] and accelerated with CUDA 8.1 [3] and cuDNN 5.1[4] on Ubuntu 14.04 LTS OS. The networks were trained on a system with 3×GTX TitanX GPU each with 12GB RAM, 2×Intel Xeon E5 2620 v3 processor and 176 GB of RAM. The codes used for implementing the framework is available at `https://github.com/kaustuv293/Tool-Detection`. The time taken for training and testing is also summarized in Tab. 1. We have trained the CNN models (BL1, BL3 and BL5) for 2,000 epochs and the LSTM for adjuncted models (BL2, BL4 and proposed framework) for 2,000 epochs.

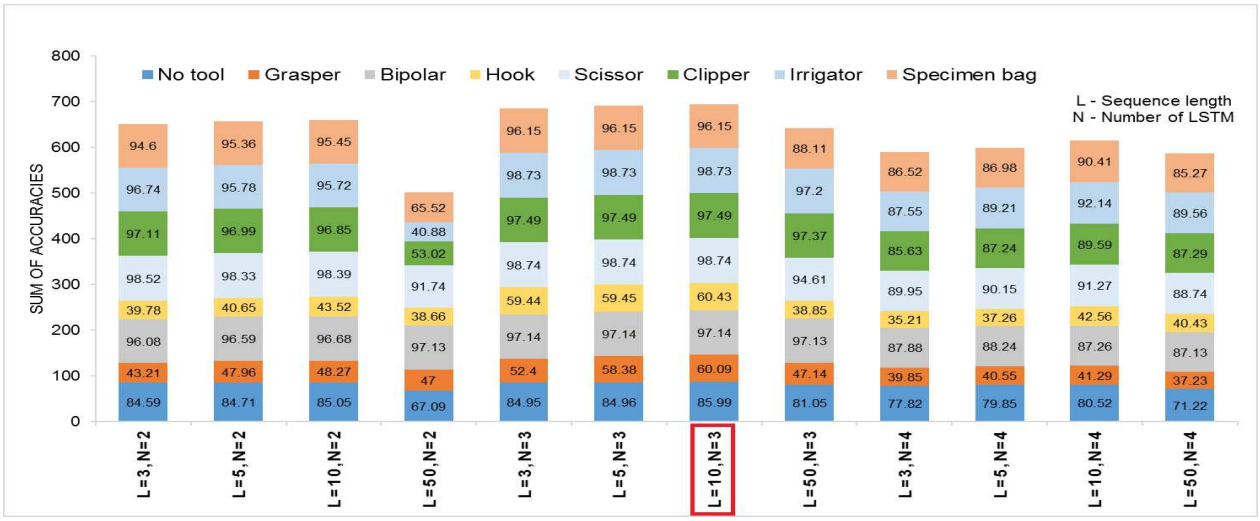
[2]http://torch.ch/

[3]https://developer.nvidia.com/cuda-downloads

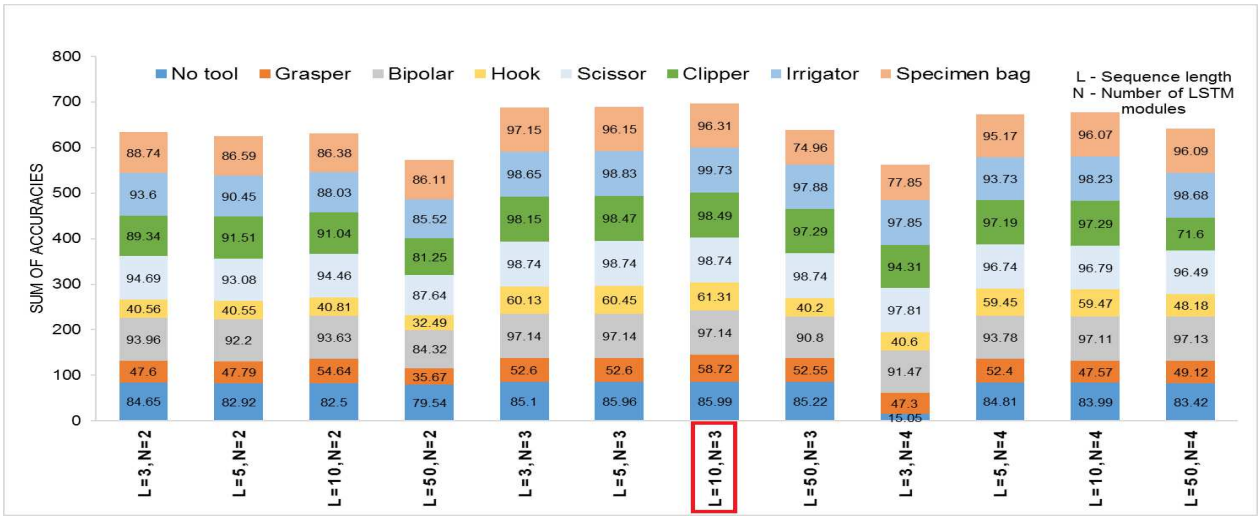
[4]https://developer.nvidia.com/cudnn

5.6. Results

With baselines **BL2, BL4** and proposed framework, we experimented on the depth of the stacked LSTM network and the length of the sequence fed into the LSTM. We have evaluated the performance for the baselines with 2, 3 and 4 stacked LSTM networks and sequence lengths of 3, 5, 10, and 50. The performance of **BL2** with the different depths of the network and sequence length is shown in Fig. 9(a). Performance of **BL4** is shown in Fig. 9(b). Performance of proposed framework is shown in Fig. 9(c). Performance comparison of **BL1, BL3** and **BL5** with the best performing configuration of **BL2** and **BL4** and the proposed method is shown in Fig. 8. The error in the prediction for each frame of one of the test video is graphically illustrated in Fig. 10.



(a) Performance of BL2



(b) Performance of BL4

(c) Performance of proposed method

Figure 9. Performance of **BL2, BL4** and proposed method with different depth and sequence lengths. The best performing configuration is marked with a red box in each case.

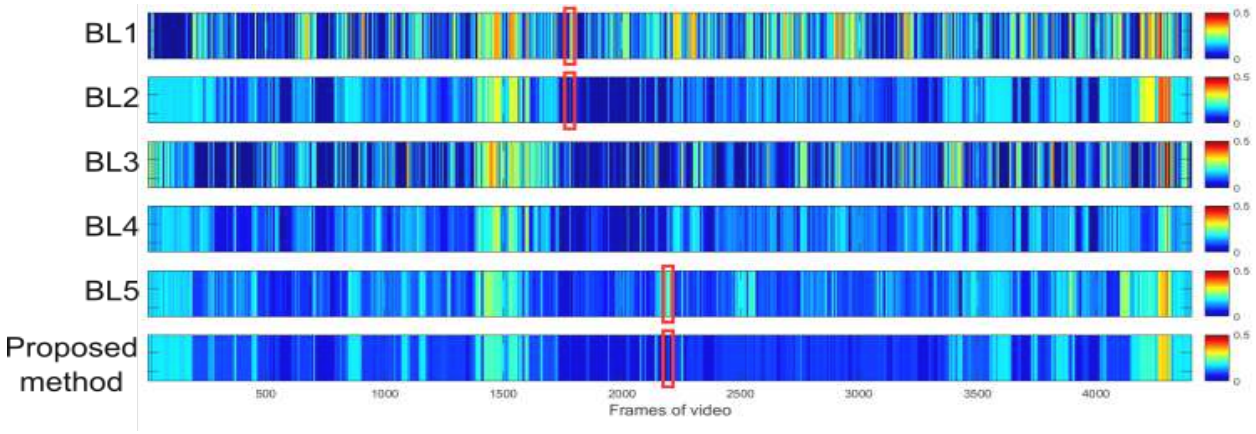
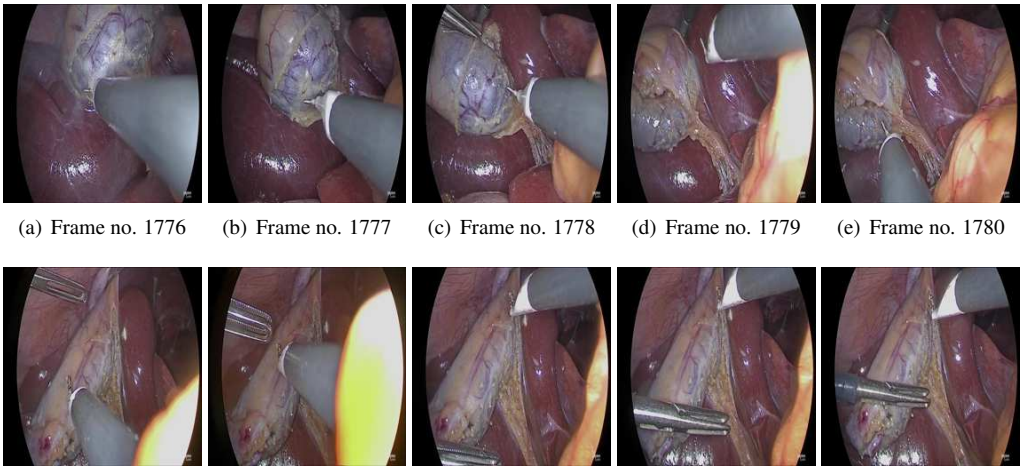


Figure 10. Error in the prediction for each frame on one of the test videos no. 11.



(a) Frame no. 1776 (b) Frame no. 1777 (c) Frame no. 1778 (d) Frame no. 1779 (e) Frame no. 1780

(f) Frame no. 2273 (g) Frame no. 2274 (h) Frame no. 2275 (i) Frame no. 2276 (j) Frame no. 2277

Figure 11. Transitions in the video causing causes errors in tool detection as in observed in (a)-(e) corresponding to BL1 and BL2 in Fig. 10 and for (f)-(j) corresponding to BL5 and Proposed method in Fig. 10.

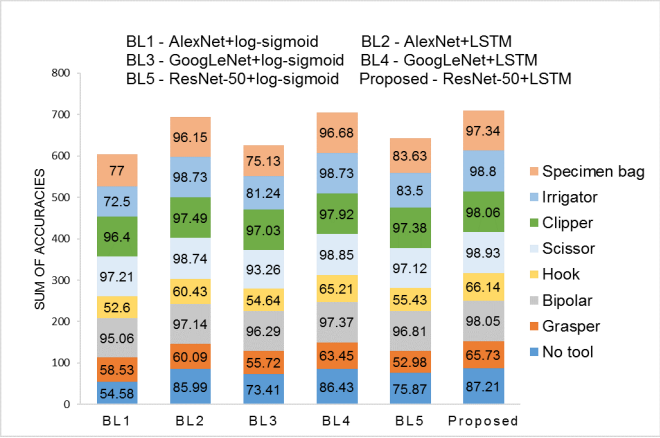


Figure 8. Performance of **BL1, BL2, BL3, BL4, BL5** and proposed method.

Video demonstrating the predictions on a test video clip can be viewed at `https://youtu.be/IyXc5F78ZU4`.

6. Discussion

As can be seen from Fig. 6(a), the occurrence of various tools in the videos of the dataset is heavily unbalanced. Out of the seven tools and their combinations, hook and hook-grasper combination occurs the most number of times. Whereas, scissors and bipolar occur scarcely. Training the network on such an unbalanced data results in the network getting biased to the tool or tool combination having maximum rate of occurrence. When trained of the balanced training data graphically illustrated in Fig. 6(b) the network learns to detect tool presence with significant accuracy as presented in Sec. 5.6.

It is observed that learning of temporal connectionism decreased the local error in detection of tool presence. The

red boxes in Fig. 10 in BL1 and BL5 are centered at frames 1,778 (Fig. 11(c)) and 2,275 (Fig. 11(h)) respectively. In these frames, either a new tool is being is introduced or there is a significant shift in the position of the tools. These transitions result in erroneous detection. With the introduction of temporal learning in BL2 and proposed method, it can been seen in that the error decreased considerably for these frames. The sequence of frames preceding and succeeding the transitional frames are shown in Fig. 11 for more clarity.

The proposed method can be broken down into two stages. Where in stage one, a CNN is trained to detect tool presence in the frames of a surgical video. In stage two, the features learned by the CNN in the first stage is used to learn a temporal model using LSTM for tool presence detection. It is observed that the accuracy of detection for the stage two (BL2, BL4 and proposed method) surpasses that of stage one(BL1, BL3 and BL5). This can be attributed to the fact that in stage one, tool detection is performed on individual frames of the video without considering the information contained in the previous frames which could lead to incorrect out of context detections. On the other hand, the temporal learning in the stage two processes a sequence of frames for detecting tool presence in one frame of the video thereby decreasing chances of false detections.

7. Conclusion

We have presented a framework for automated tool presence detection in laparoscopy videos which can aid in surgical workflow modeling. In the proposed method, a deep LSTM network presented in Sec. 4.2 is trained to detect tool presence in surgical videos using spatial visual features extracted from the frames using a CNN trained on ILSVRC [3] followed by fine-tuning on a publicly available laparoscopy video dataset. The performance of the proposed framework is experimentally verified by comparing with a set of baselines. It is observed that our framework, and specifically the proposed method using deep residual spatial attributes of the images and learning with temporal connectionism with LSTM outperforms the different baselines in terms of detecting the tool presence end with high accuracy of {87.21%, 65.73%, 98.02%, 66.14%, 98.93%, 98.06%, 98.8% and 97.34%}for the tools classes {No tool, Grasper, Bipolar, Hook, Scissors, Clipper, Irrigator, Specimen bag} and average tool detection accuracy of 88.75%.

References

[1] J. Bergstra, G. Desjardins, P. Lamblin, and Y. Bengio. Quadratic polynomials learn better image features. Technical report, 1337, Department of Informatics and Optical Research, Universit e de Montr al, 2009.

[2] A. Casals, J. Amat, and E. Laporte. Automatic guidance of an assistant robot in laparoscopic surgery. In *IEEE Int. Conf. Robo. Autom.*, volume 1, pages 895–900, 1996.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recog.*, 2009.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recog.*, pages 770–778, 2016.

[5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Info. Process. Sys.*, pages 1097–1105, 2012.

[7] D. Krueger and R. Memisevic. Regularizing rnns by stabilizing activations. In *Proc. Int. Conf. Learning Rep.*, 2016.

[8] A. Krupa, J. Gangloff, C. Doignon, M. F. de Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescaux. Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing. *IEEE Trans. Robotics Autom.*, 19(5):842–853, 2003.

[9] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[10] F. Miyawaki, T. Tsunoi, H. Namiki, T. Yaginuma, K. Yoshimitsu, D. Hashimoto, and Y. Fukui. Development of automatic acquisition system of surgical-instrument informantion in endoscopic and laparoscopic surgey. In *IEEE Conf. Indu. Elec. App.*, pages 3058–3063, 2009.

[11] A. Reiter, P. K. Allen, and T. Zhao. Articulated surgical tool detection using virtually-rendered templates. 2012.

[12] J. Ryu, J. Choi, and H. C. Kim. Endoscopic vision based tracking of multiple surgical instruments in robot-assisted surgery. In *Int. Conf. Control, Autom. Sys.*, pages 2195–2198, 2012.

[13] S. Speidel, E. Kuhn, S. Bodenstedt, S. Röhl, H. Kenngott, B. Müller-Stich, and R. Dillmann. Visual tracking of da vinci instruments for laparoscopic surgery. In *SPIE Med. Imag.*, pages 903608–903608, 2014.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recog.*, pages 1–9, 2015.

[15] R. Sznitman, C. Becker, and P. Fua. Fast part-based classification for instrument detection in minimally invasive surgery. In *Int. Conf. Medical Image Comput. Comp. Assist. Interv.*, pages 692–699, 2014.

[16] O. Tonet, R. U. Thoranaghatte, G. Megali, and P. Dario. Tracking endoscopic instruments without a localizer: a shape-analysis-based approach. *Comp. Aided Surgery*, 12(1):35–42, 2007.

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