Automatic Detection of Surgical Phases in Laparoscopic Videos

Babak Namazi  
Department of Electrical Engineering  
University of Texas at Arlington  
Arlington, TX 76019  
babak.namazi@mavs.uta.edu

Ganesh Sankaranarayanan  
Department of Surgery  
Baylor University Medical Center  
Dallas, TX 75246  
ganesh.sankaranarayanan@BSWHealth.org

Venkat Devarajan  
Department of Electrical Engineering  
University of Texas at Arlington  
Arlington, TX 76019  
venkat@uta.edu

Abstract

The assessment and evaluation of surgical skills require a considerable amount of time and effort. Currently, the assessment is accomplished by either observing a recorded video the surgery in the case of laparoscopic procedures or watching it in real-time in an operating room in the case of open surgical procedures. Given the limited time available to experts, knowing the surgical work-flow can expedite the process of assessment by helping the surgeons to focus on the most important parts of each surgical phase separately. In this paper, we propose a novel method to identify the surgical phases using recorded videos (the data set) of a laparoscopic procedure. A deep learning (DL) Convolutional Neural Network (CNN) is used followed by a recurrent neural network (RNN) to consider both spatial and temporal information to classify the video frames. In order to evaluate the resulting DL model, we used the publicly available cholec80 dataset which contains 80 videos of laparoscopic cholecystectomy procedure. Our experimental results show significant improvement in the identification of surgical phases over existing methods in both real-time and off-line modes.

1 Introduction

Laparoscopic surgery, also known as minimally invasive surgery (MIS), is increasingly a common and preferred technique over open surgery due to their better quality of life outcomes, less pain and faster recovery time[1]. In a laparoscopic procedure, the internal organs are accessed through long slender instruments inserted via small incisions on the abdomen. A specialized camera tool, which is also placed through a separate incision is used to visualize the surgical field. Despite its benefits, there are several challenges in performing a laparoscopic surgery such as limited degrees of freedom of movement, limited field of view and lack of depth perception etc. [2]. Therefore, it takes a considerable amount of training to master laparoscopic surgical skills.

Routine assessment of surgical skills, on the other hand, requires expert supervision in the operating room (OR) while the surgery is being performed. Alternatively, in minimally invasive surgeries, the evaluation can be accomplished using the videos recorded from the procedure[3]. However, this still requires a significant amount of the surgeon’s time and is subject to human bias. Separating the videos into the corresponding surgical phases can help the surgeons find and evaluate the video...
Figure 1: Illustration of the challenges in detection surgical workflow using still images. a) examples of blurry images and lack of focus in camera, b) Using the same tool (Bipolar) in two different phases, c) absence of surgical instruments in different steps

parts faster. However, the most productive approach results, if the segmented parts can be used in an automated system to perform the evaluation using objective measures. The final output videos will have all the phases identified and therefore can ease access to the information stored in video databases during the training of the surgical residents.

Despite the availability of the recorded videos, automatic detection of surgical steps using still frames from videos is quite challenging because, the rapid movement of the camera causes blurry views and low-quality images. Also, the camera is not always focused on the scene. This is shown in figure 1a. Thus, using still frames for classification is not sufficient for having a reasonable performance.

Traditionally, identifying various surgical phases was accomplished using the signals from the tools being used during the procedure[4] or from the manually annotated videos[5] with features such tool, anatomical structure, and surgical tasks[6]. However, designing such hand-crafted features is time-consuming and is not optimal. These shortcomings have been recently addressed using DL methods and CNN in particular.

CNNs are special types of deep artificial neural networks designed to extract the high-level visual features in order to perform various tasks on images and videos. Recently, CNNs have shown promising performance in computer vision tasks such as image and video classification, object detection and segmentation, action recognition, etc[7, 8]. Therefore, deep CNNs have been the preferred choice in detecting surgical phases in the recent years.

Endonet[9] was the first model that used visual features using a CNN architecture called Alexnet[7]. In this work, the CNN is trained for extracting features for simultaneously identifying surgical instruments and phases. The output of the tool detection system is concatenated with the features and used for phase detection classifier. The result of the last layer (after training) is then used as the input to a Hierarchical Hidden Markov Model (HHMM) to incorporate the temporal co-occurrence of the features into the decision-making system. [10] used a Gaussian distribution fitting model to deal with the correlation of the video frames. They further used random forest classifier to improve the detection accuracy.

In their other work, Twinanda et al.[11] replaced the HMM model with a long short-term memory (LSTM) model. Although using tool information showed some improvement in some metrics, the accuracy remains the same when using LSTM. The reason is that not all of the surgical phases have a correlation with tools being used. This is shown in image 1a and c. As a result, using just the tool information may not give a significantly better accuracy.

In order to train a CNN with LSTM in an end-to-end fashion, EndoRCN[12] used three consecutive frames as the input sequence to the recurrent network. Similarly, SVRCN[13] used a short sequence of frames for simultaneously training CNN and LSTM. Although this model takes the benefits of the end-to-end training and the smoother results due the short window of the frames, it misses the long-term relationship between the video phases.

The other approach for doing phase detection in surgical videos is to use a recently developed architecture called two-stream networks[14]. In this model, one stream is taking the still images as input for extracting the visual features, while the other one uses a short stack of motion feature images, such as optical flow to take the temporal information into account. While these models are extremely useful in some video processing tasks such as activity recognition, they are complicated to set up and cannot be used in real-time application.

In this paper, we designed two separate architectures for real-time and offline models. In both models,
in addition to the visual features, the frame number is added as a separate feature, before the fully connected layer of the CNN. This method helps the CNN know the actual position of the frame with respect to the other frames of the video. In the offline model, a temporal smoothing method is applied to a short window of the frames. The output of both models is sent to a separate LSTM network for the final decision making. The proposed model is tested with a dataset of cholecystectomy procedure and showed the state-of-the-art results in detecting surgical workflow. The main contributions of the paper are as follows:

- Having separate architectures for online and offline detection of surgical phases
- Using both short-term and long-term correlations of features in the videos
- Using frame number (time) along with the other high level visual features to improve phase detection accuracy
- Addressing the class imbalance problem of surgical phases
- Dealing with the over-fitting issue (to be explained later) by separating the training sets of the CNN and LSTM models

The remainder of the paper is organized as follows: in section 2 the methodology is described in detail. Section 3 describes the setting used while in section 4, the proposed model is evaluated and the results are compared to the recent comparable methods. The final section concludes the paper and some suggestions for future work are stated.

2 Methodology

In this section, we describe the model designed for segmenting surgical videos according to their procedural phases. The proposed model consists of two architectures for online and offline cases. Both architectures rely on a CNN for extracting spatial features in each frame of the videos and an RNN for temporal dependencies of the subsequent frames.

2.1 Convolutional Neural Networks

Extracting the high level features of images is at the heart of most computer vision tasks. Similarly, having highly informative features is important for identifying surgical tasks in a video. Recently, CNNs have shown promising results in several tasks including image classification, object detection, semantic segmentation, etc.

Each CNN consists of a number of convolutional kernels followed by a pooling mechanism to reduce the dimensionality of the input to a feature map. A fully connected network is then applied to the resulting feature maps to reach the final output of the model.

Since phase detection is a multi-class problem, we use a softmax cross-entropy loss function to find the probability of each class. The class with the maximum probability is selected as the CNN prediction of the current phase, for sending to the RNN to deal with the temporal correlations of the frames.

2.2 Imbalance

Class imbalance has been a well-known problem in machine learning for decades[15]. Skewed datasets result in higher tendency towards the higher frequency classes and their lower overall accuracy. This issue exists in surgical videos as well. As can be seen in table[9], the duration of each of the phases of cholecystectomy procedure varies significantly in each video. In order to address this class imbalance problem, we used an up-sampling method, which interpolates the video frames for less frequent classes. The resulting dataset has a uniform distribution among all classes.

2.3 Over-fitting

Recent CNN architecture such as residual networks[16] and Inception[17][8] are designed to classify a large number of images into 1000 classes. Each model consists of tens of layers, each with thousands
Table 1: Duration of each phase in cholecystectomy procedure

<table>
<thead>
<tr>
<th>Phase</th>
<th>duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation</td>
<td>125 ± 95</td>
</tr>
<tr>
<td>Calot Triangle Dissection</td>
<td>954 ± 538</td>
</tr>
<tr>
<td>Clipping and Cutting</td>
<td>168 ± 162</td>
</tr>
<tr>
<td>Gallbladder Dissection</td>
<td>857 ± 551</td>
</tr>
<tr>
<td>Gallbladder Packing</td>
<td>98 ± 53</td>
</tr>
<tr>
<td>Cleaning and Coagulation</td>
<td>178 ± 166</td>
</tr>
<tr>
<td>Gallbladder Packing</td>
<td>83 ± 56</td>
</tr>
</tbody>
</table>

of parameters to train. Due to the large number of parameters, these architectures are prone to significant over-fitting error. Although surgical videos are taken in high resolution (25 or 30 frames per second), the diversity in the images is not sufficient for high generalization in a deep CNN; hence perfect accuracy during training and large error during inference occur. While the use of various regularization methods such as L1 and L2 and dropout, reduces the generalization error, there is still a big gap between the training and test accuracy in surgical videos. In order to deal with this issue, we split the training set into two smaller sets for training CNN and LSTM separately. This ensured the input sequence during training RNN is similar to the sequence at the test time. We believe this approach helps the RNN learn the input sequence better.

2.4 Temporal correlations

While convolutional neural networks are suitable for extracting spatial visual information in still images, they don’t take temporal dependencies into account. As can be seen in figure[1] because of the rapid camera or surgical instrument movements, not all of the images are clear enough to be used with CNN. We deal with this problem using a temporal smoothing method, operating on a short window, during test time. This was accomplished by applying a fixed size moving average window on the last layer’s features or the final prediction. However, due to the high correlation of neighboring frames in a video, an error can be propagated to a number of frames in a row. Thus, using long-term temporal dependencies is crucial in frame level classification in a video. We used Long Short-Term Memory (LSTM)[18], which is a type of recurrent neural network suitable for long sequences. LSTMs have been used for time-series modeling such as sequence classification and sequence labeling. In order to use the full potential of the LSTM model, we apply it to the entire video. However, due to memory constraint, it’s not possible to train both RNN and CNN in an end-to-end fashion.

For the input of the recurrent model, one approach is to use the extracted features from the last convolutional layer of the convolutional architecture. While these features have a good representation of the visual features, it is more complicated to train an RNN with our large input size. On the other hand, the number of features in the last layer varies for each architecture. Alternatively, we can use the outputs of the CNN, which are the probabilities of all classes, or the final predictions. We used the prediction for both online and offline model, which enabled us to have a shallower RNN without sacrificing accuracy.

2.5 Frame number

Since the duration of the phases and their order are known to us, we can use the frame number to improve the performance of the CNN. The frame number or the time of each frame is used as an additional feature in the last layer of our convolutional network. The fused features were then be used with the fully connected and softmax layers. This way, the CNN is aware of the relative position of each frame during training.

2.6 Offline architecture

During the offline mode, we assume that the model has access to all the frames before and after the current one. Also, the total number of frames in a video is given. For training the CNN, the frame number is divided by the total size of the video (normalized) and is concatenated with the features of the last convolutional layer and right before the fully connected layer. The output of a separate
Figure 2: The block diagram of the proposed model in offline mode. The images are sent in a fixed size batch (here 20), through a CNN. The red cube on top of the feature column is the frame number. The results of the fully connected (FC) layer for all the batches are smoothed and used in the LSTM model, to get the final output, which is the probability of each of the surgical phases.

training set (the final prediction of the CNN) is then smoothed by averaging over a fixed size window. The results are used for training a bi-directional LSTM model. A Bi-RNN consists of two RNNs, one in the forward direction and the other in the backward direction. Thus, the bi-directional LSTM can use the information of the frames from the past and future. The block diagram of the model is shown in the figure 2.6.

2.7 Online architecture

In the online detection of surgical workflow, the only information available is the features from the previous frames. Therefore, the temporal smoothing method cannot be used. Therefore, a uni-directional LSTM is applied to the output of the convolutional neural network. Since the size of the video is unknown at the processing time, the time-stamp feature is just the frame number concatenated with feature map of CNN.

3 experimental setups

We used cholec80 dataset[9] to evaluate our model. The dataset contains 80 videos from cholecystectomy procedure performed by 13 surgeons. All the videos were labeled with the tools and seven surgical phases. We used 40 videos for training the CNN, 20 for training LSTM model and 20 for testing. The training set is balanced with 50000 samples per class. We used the Inception V1 model for extracting the predictions. All the images were resized to 224 by 224, which is the default input size of the inception v1 model. The images first went through a number of online data augmentations, such as flipping left to right and up-down, cropping and resizing, color distortion, random rotation, etc. The model was initialized with the pretrained weights from the Imagenet dataset. For better generalization results, we used l2 regularization (weight decay) and dropout. The window size for our temporal smoothing model in offline mode was set to 20.

The RNN model consisted of two LSTM layers with 128 and 16 units for both online and offline phases (bi-directional LSTM for offline mode). Since the number of videos for training LSTM is not sufficient, we used data augmentation for the input sequences too. The predictions were saved every 20 frames and for each sequence of predictions, 80 percent of the frames were picked randomly at each step during training. The resulting sequences were of slightly different sizes (and phase durations) from the original videos. A small random noise was added to the input as well. Since the videos have different sizes, we padded the input sequences with a new class to have equal input size for the dynamic RNN implementation. The same padding was added for Bi-LSTM and in the backward direction.

All the experiments were performed with an Nvidia GTX 1080 Ti and using the Tensorflow
Table 2: The accuracy of the CNN models

<table>
<thead>
<tr>
<th>CNN</th>
<th>time-stamped on-line</th>
<th>time-stamped off-line</th>
<th>average Smoothing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>78.2</td>
<td>79.97</td>
<td>80.68</td>
</tr>
</tbody>
</table>

The total processing time for the online model was below 0.02 seconds per image, which is less than the requirements of real-time applications.

4 Results

In this section, the results from various experiments are demonstrated. The performance is then compared with the previous work.

Table 2 summarizes the results of the convolutional neural networks. It shows that having temporal information improves the performance of the feature extractor model. The difference between the results of the CNN after time-stamping in online and offline mode is the result of the normalization in the offline mode, whereas during on-line detection, the absolute number is used which takes longer to converge and has less accuracy. Since the input to our LSTM network is the prediction from the CNN, a small increase in accuracy makes a big difference in the output of the model. This can be seen in the last column of the table which is the accuracy after applying the temporal smoothing method.

Since the testing set is not balanced, we need to check performance of the models, for all of the phases separately, using precision and recall metrics.

The results of the experiments for the offline and online cases are shown in tables 3 and 4 respectively. The f1 score measured in the tables is the harmonic mean of precision and recall. It can be seen that cleaning and coagulation has the least accuracy in both online and offline modes. This is probably because the order of the last two phases were reversed in some videos; the cleaning can happen before the complete retraction of the gallbladder. On the other hand, Calot triangle dissection and gallbladder dissection have the highest f1-score in both modes. The reason is likely the higher duration of these phases according to Table 1. The accuracy in online mode is 90.8 whereas in offline mode the accuracy is 96.5. The improvement in the accuracy is the result of having temporal smoothing and bi-directional LSTM.

In order to have better visualization of the per-class results, the confusion matrix is shown in

Table 3: Precision, recall and f1 for offline mode

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation</td>
<td>0.91</td>
<td>0.88</td>
<td>0.89</td>
<td>2162</td>
</tr>
<tr>
<td>Calot Triangle Dissection</td>
<td>0.97</td>
<td>0.99</td>
<td>0.98</td>
<td>17456</td>
</tr>
<tr>
<td>Clipping and Cutting</td>
<td>0.97</td>
<td>0.95</td>
<td>0.96</td>
<td>2951</td>
</tr>
<tr>
<td>Gallbladder Dissection</td>
<td>0.99</td>
<td>0.97</td>
<td>0.98</td>
<td>19144</td>
</tr>
<tr>
<td>Gallbladder Packing</td>
<td>0.98</td>
<td>0.97</td>
<td>0.97</td>
<td>2089</td>
</tr>
<tr>
<td>Gallbladder Retraction</td>
<td>0.93</td>
<td>0.90</td>
<td>0.91</td>
<td>1530</td>
</tr>
<tr>
<td>Cleaning and Coagulation</td>
<td>0.85</td>
<td>0.86</td>
<td>0.86</td>
<td>2293</td>
</tr>
<tr>
<td><strong>avg/total</strong></td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>47625</td>
</tr>
</tbody>
</table>

Table 4: precision, recall and f1 score for online mode

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation</td>
<td>0.88</td>
<td>0.90</td>
<td>0.89</td>
<td>2162</td>
</tr>
<tr>
<td>Calot Triangle Dissection</td>
<td>0.94</td>
<td>0.97</td>
<td>0.96</td>
<td>17456</td>
</tr>
<tr>
<td>Clipping and Cutting</td>
<td>0.72</td>
<td>0.81</td>
<td>0.76</td>
<td>2951</td>
</tr>
<tr>
<td>Gallbladder Dissection</td>
<td>0.96</td>
<td>0.92</td>
<td>0.94</td>
<td>19144</td>
</tr>
<tr>
<td>Gallbladder Packing</td>
<td>0.87</td>
<td>0.84</td>
<td>0.85</td>
<td>2089</td>
</tr>
<tr>
<td>Gallbladder Retraction</td>
<td>0.82</td>
<td>0.65</td>
<td>0.72</td>
<td>1530</td>
</tr>
<tr>
<td>Cleaning and Coagulation</td>
<td>0.65</td>
<td>0.74</td>
<td>0.69</td>
<td>2293</td>
</tr>
<tr>
<td><strong>avg/total</strong></td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>47625</td>
</tr>
</tbody>
</table>

Class 7 is the padded class. It can be seen that most of the errors are happening in the
neighboring classes. Again, the higher tendency towards P4 (Gallbladder Dissection) is because of the longer duration of the phase.

Table 5 compares our work with the recent comparable papers. The high accuracy in the proposed model in offline mode shows that considering both short-term and long-term dependencies of the surgical video frames can significantly improve the performance. Moreover, using the frame number information helps the entire model use the time of the frame being classified. Having the training sets separated for the CNN and LSTM and diverse data augmentation techniques are the main reasons for the improvement in the performance of the proposed model compared to the existing methods.

Table 5: Overall Accuracy

<table>
<thead>
<tr>
<th>model</th>
<th>online</th>
<th>offline</th>
</tr>
</thead>
<tbody>
<tr>
<td>endonet (HMM)</td>
<td>82.0</td>
<td>91.0</td>
</tr>
<tr>
<td>endonet (LSTM)</td>
<td>88.6</td>
<td>92.2</td>
</tr>
<tr>
<td>SVRCN[13]</td>
<td>N/A</td>
<td>90.7</td>
</tr>
<tr>
<td>ours</td>
<td><strong>90.8</strong></td>
<td><strong>96.3</strong></td>
</tr>
</tbody>
</table>

5 Conclusion and Future Direction

A novel model is proposed for segmenting surgical videos into surgical phases. The model takes advantage of the spatial and temporal features to classify all of the surgical videos. A CNN followed by a temporal smoothing extracts useful information to send to an LSTM model for getting the final output of detected phases. In the online mode, the CNN is aware of the frame time and is able to do the classification in real-time. There are few possible future improvements to our method. The temporal smoothing method applied on the final prediction of the CNN, is blind to the actual visual features of the frames. We are planning on replacing it with a recurrent neural network that can capture short-term correlations. Additionally, the class imbalance in the LSTM output will be addressed by using class weights.

6 Acknowledgement

This work was supported by a SEEGER grant from Baylor University Medical Center, Dallas, TX, USA.

References


