Hard Frame Detection and Online Mapping for Surgical Phase Recognition

Fangqiu Yi and Tingting Jiang(✉)

NELVT, Department of Computer Science, Peking University, Beijing, China
{chinayi,ttjiang}@pku.edu.cn

Abstract. Surgical phase recognition is an important topic of Computer Assisted Surgery (CAS) systems. In the complicated surgical procedures, there are lots of hard frames that have indistinguishable visual features but are assigned with different labels. Prior works try to classify hard frames along with other simple frames indiscriminately, which causes various problems. Different from previous approaches, we take hard frames as mislabeled samples and find them in the training set via data cleansing strategy. Then, we propose an Online Hard Frame Mapper (OHFM) to handle the detected hard frames separately. We evaluate our solution on the M2CAI16 Workflow Challenge dataset and the Cholec80 dataset and achieve superior results. (The code is available at https://github.com/ChinaYi/miccai19).

Keywords: Surgical phase recognition · Data cleansing · Deep learning

1 Introduction

Computer-Assisted Surgery (CAS) systems are crucial in the development of modern surgery. Surgical phase recognition is an important topic of CAS systems because it offers solutions to numerous demands of the modern operating room(OR). For instance, such recognition is an important component to develop context-aware systems to monitor surgical processes [3], schedule surgeons [1] and enhance coordination among surgical teams [10]. The surgical phase recognition can be performed online or offline. The online surgical phase recognition is more challenging than offline since we are not allowed to use the information of future frames. However, online surgical phase recognition is more suitable for practical application, since the online recognition can support decision making during the surgery, especially for junior surgeons. This paper works on the online surgical phase recognition task.

Previous online surgical phase recognition approaches can be classified into two categories. The first one is dedicated to extracting discriminative visual

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-32254-0_50) contains supplementary material, which is available to authorized users.

D. Shen et al. (Eds.): MICCAI 2019, LNCS 11768, pp. 449–457, 2019.
https://doi.org/10.1007/978-3-030-32254-0_50

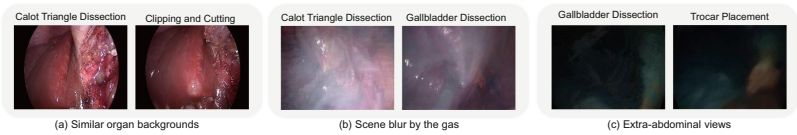

Fig. 1. Illustration of various hard frames in M2CAI16 Workflow Challenge. The text on the top of each frame indicates which phase it belongs to.

features to train a frame-wise classifier without utilizing temporal information, while the second one tries to combine temporal information in different manners. However, we observe that both approaches suffer from the existence of hard frames. As demonstrated in Fig. 1(a), (b) and (c), there are three types of hard frames. For hard frames of the same type, they have indistinguishable visual features but are assigned with different labels during the annotation. Owing to the visual similarity of hard frames, frame-wise methods perform poorly, and an example result is shown in Fig. 2(a). Combining temporal information can effectively help to classify hard frames, however, with the interference of hard frames, the captured temporal information may be heavily disturbed, resulting in additional errors, as shown in the Fig. 2(b). Moreover, the detection of spikes is not a trivial thing for online surgical workflow segmentation, resulting in severe over-segmentation problem.

One solution to address the challenge is to detect hard frames and handle them separately, which can benefit both training and testing. To be specific, for training, the negative impact of hard frames will be minimized if hard frames can be removed. For testing, the detected hard frames can be labeled as an additional class to separate from other simple frames, which can alleviate the disruption on the temporal structure. Meanwhile, the detected hard frames can be further treated by an online rectifying mechanism.

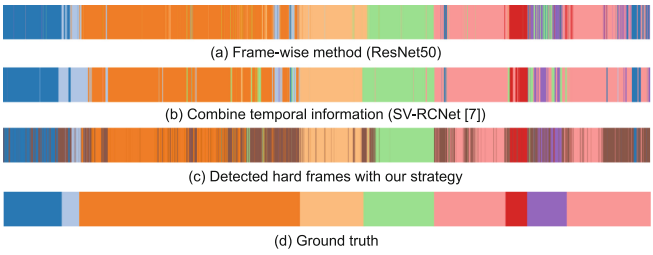

Fig. 2. Visualization of classification results on one test surgery in M2CAI16 Workflow Challenge. (a) Frame-wise method. (ResNet50 is used for illustration.) (b) Method combining temporal information. (SV-RCNet w/o post-processing [7] is used for illustration.) (c) Detected hard frames with our strategy, shown as the brown ribbon. (d) Ground truth labels.

Motivated by the above analyses, we propose a three-step approach to deal with the problems of hard frames. Since only the beginning and the end of a phase are manually annotated, hard frames that within the phase are automatically labeled. Therefore, we first take hard frames as mislabeled samples and employ a data cleansing strategy based on model predictions to find out hard frame samples in the training set. Next, these hard frames are labeled as an additional class to separate from other simple frames, and a classifier is trained to carry out both detection task for hard frames and phase recognition task for simple frames during the test, and example results are shown in Fig. 2(c). Finally, for the detected hard frames, we propose an Online Hard Frame Mapper (OHFM) to map them to corresponding phases. We extensively evaluate our solution on the M2CAI16 Workflow Challenge dataset and Cholec80 dataset. Our main contributions are summarized as follows. (1) For the first time, we explicitly raise the issues of hard frames and propose a novel solution for surgical phase recognition. (2) Our proposed solution achieves superior results on two benchmark datasets.

2 Related Work

Surgical Phase Recognition. Numerous approaches have been proposed to perform the surgical phase recognition task. Early studies use a frame-wise classifier to tackle videos frame-by-frame without using temporal information. These works focus on extracting discriminative visual features, such as various handcrafted features [2,8] or deep CNN features [16]. The other type of approaches tries to combine temporal models in different manners. For example, a number of works utilize dynamic time warping [2,13], conditional random field [15], and variations of Hidden Markov Model (HMM) [9,12] to enforce temporal constraints to the output results. Jin et al. [7] train an end-to-end CNN-RNN model to encode both spatial and temporal information, which is referred to as SV-RCNet. However, the captured temporal information may contain noises caused by hard frames, leading to unreliable classification results. Furthermore, some works apply post-processing strategy to rectify the results, such as PKI [7], avg-smoothing [4]. However, the improvement by post-processing may highly depend on the hyper-parameters, and the generalization ability is limited.

Data Cleansing. Many methods have been proposed to cleanse training sets, with different degrees of success [5]. Some methods detect mislabeled instances with measures like the classification confidence [14] or the model complexity [6]. However, these methods are applicable to the condition where mislabeled samples are only a small part of the training set. While in surgical videos, the ratio of hard frames may be relatively large according to our experiment. Another type of methods relies on the predictions of classifiers [11]. They use a K-fold cross-validation scheme to obtain the predictions on every validation set, and then determine from the validation results whether a sample is mislabeled or not. In this paper, we adopt the second type of methods, but we take the detected samples as an additional class rather than ignoring it.

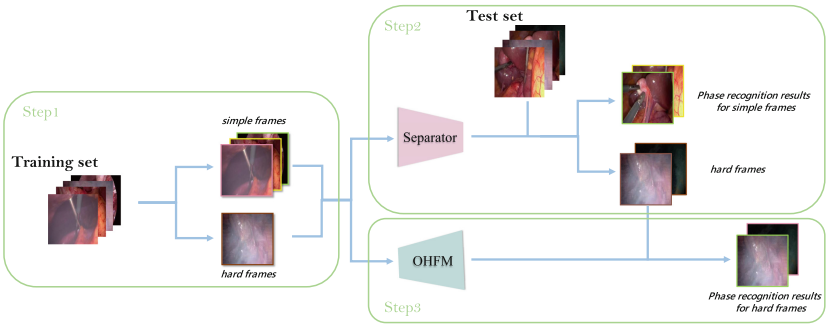

Fig. 3. The overview of our three-step solution for phase recognition task.

3 Methodology

The input of online phase recognition task is a video sequence $X = \{x_1, x_2, ..., x_m\}$ with N_p phases, the output is the corresponding phases $\{c_1, c_2, ..., c_m\}$ for each frame in the video sequence, where $c \in \{1, 2, ..., N_p\}$. Our solution consists of three main steps to tackle the phase recognition task, as illustrated in Fig. 3, and will be described in detail in following sections.

3.1 Data Cleansing for Training

Motivated by the observations that frame-wise classifiers often make mistakes on hard frames, we use a model-prediction based data cleansing strategy [5] which has two stages. The first stage consists of using a K-fold cross-validation scheme. Specifically, we randomly partition the training videos into K groups of equal size. Each time, a single group is retained for validation, and the remaining $K-1$ groups are used to train a frame-wise classifier. In our experiment, we use *ResNet50* as our classifier, but it can be replaced with any frame-wise classification method. The second stage is to determine from the validation results whether a sample is hard or not. We simply take samples that are misclassified as hard frames.

3.2 Hard Frame Detection for Testing

We take hard frames in the training set as an additional class and train another ResNet50 classifier with $(N_p + 1)$ classes, which is referred to as "*Separator*" for further discussion. *Separator* carries out the detection task for hard frames and the recognition task for simple frames simultaneously by outputting the class label $\hat{c}$ for the input frame x, where $\hat{c} \in \{0, 1, 2, ..., N_p\}$. To be specific, $\hat{c} \in \{1, 2, ..., N_p\}$ represents the phase recognition results while $\hat{c} = 0$ stands for the detected hard frame, which will be further rectified by the Online Hard Frame Mapper (OHFM).

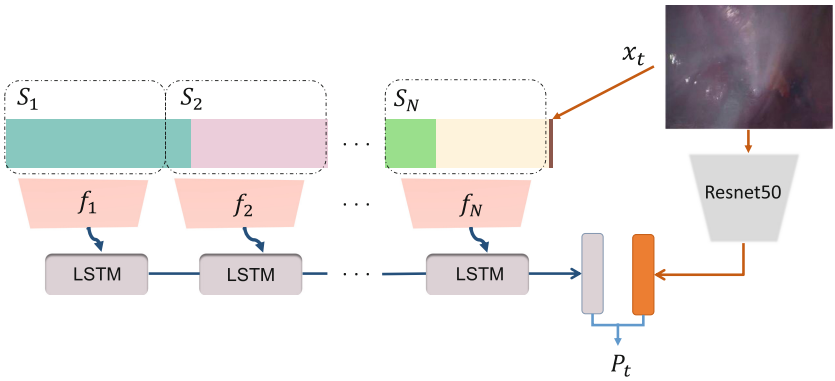

Fig. 4. Architecture of OHFM network for hard frame mapping. In LSTM branch, previous prediction sequence is split into N sub-sequences of equal length, denoted as $\{S_1, S_2, ..., S_N\}$. f_i is the extracted feature for each sub-sequence. LSTM take $\{f_i\}$ as input to obtain the prediction result for hard frame x_t. In the ResNet branch, visual features are extracted by ResNet50 to help with the mapping task.

3.3 Online Hard Frame Mapper

For a detected hard frame x_t, we proposed an Online Hard Frame Mapper (OHFM) to map it to its corresponding phase with two branches. The LSTM branch is designed for utilizing the classification sequence of previous frames while the ResNet branch tries to extract useful visual features of x_t. The architecture of OHFM is illustrated in Fig. 4.

LSTM Branch. Previous predictions are very helpful for us to map hard frame samples to its correct phase. Suppose $C_{t-1} = \{\hat{c}_1, \hat{c}_2, ..., \hat{c}_{t-1}\}$ is the previous prediction sequence given by the *Separator* which is described in Sect. 3.2. Note that hard frames are labeled as an additional class to be different from original phases. Therefore, the temporal interruption caused by the hard frames can be alleviated. We split sequence C_{t-1} into N sub-sequences of equal size, denoted as $\{S_1, S_2, ..., S_N\}$. For each sub-sequence S_i, two types of features are extracted. First, we find out the phase that appears most frequently and encode it with one-hot encoding, denoted as $M(S_i)$. Then, we calculate the proportion of each phase, denoted as $Pr(S_i)$. The change in proportion of phases in each bin is used to reflect the surgical procedure since the timestamp is not available in online surgical workflow segmentation. Both $M(S_i)$ and $Pr(S_i)$ are $(N_p + 1)$-dimension vectors, and then are concatenated to form the final feature f_i. The LSTM network takes $\{f_i\}$ as input. The output of the last LSTM cell is a N_p-dimension vector, denoted as P_{lb}, which represents the predicted probability that x_t belongs to the corresponding phases by the LSTM branch.

ResNet Branch. Visual features can be helpful in the condition where the LSTM branch is not confident about its predictions. Therefore, we construct a 50-layer ResNet and try to extract useful visual information to help with the mapping

task. For a hard frame x_t, the output of ResNet50 is also a N_p-dimension vector, denoted as P_{rb}, represents the predicted probability that x_t belongs to the corresponding phase by the ResNet branch.

Loss. The final probability is obtained by the weighted sum of two branches: $P_t = \alpha * P_{lb} + (1 - \alpha) * P_{rb}$, where α is the hyper-parameter. The loss function for the mapper is defined as a cross-entropy loss.

3.4 Implementation Details

Our framework is implemented with the PyTorch deep learning library, using 8 Tesla K80 GPU for acceleration. Since the phase recognition results given by the *Separator* may contain errors, we augment the training data with random noise to effectively train our OHFM. Specifically, for the phase that appears most frequently in each sub-sequence, it will be randomly changed to any other phases with the probability of 15%. We set $N = 240$ and $\alpha = 0.95$ for our experiment. The frames before $N = 240$ are labeled as the initial phase of the surgical video.

4 Experiment

4.1 Dataset

M2CAI16 Workflow Challenge. The M2CAI16 Workflow Challenge dataset contains 41 laparoscopic videos that are acquired at 25 fps of cholecystectomy procedures, and 27 of them are used for training and 14 videos are used for testing. These videos are segmented into 8 phases by experienced surgeons.

Cholec80. The Cholec80 dataset contains 80 videos of cholecystectomy surgeries performed by 13 surgeons. The dataset is divided into training set (40 videos) and testing set (40 videos). The videos are divided into 7 phases and are captured at 25 fps.

For training data preparation, the original videos are downsampled from 25 fps to 5 fps. The resolution of the frames is resized to 256×256 to save the GPU memory. All the results are reported on the full test set.

4.2 Data Cleansing Result

We set $K = 9$ (3 videos in 1 group) for M2CAI16 Workflow Challenge and $K = 10$ (4 videos in 1 group) for Cholec80. Table 1 shows the proportion of hard frames in each phase. As shown in Table 1, almost half of the frames in the *Preparation* phase are cleaned up as hard frames. This makes sense because illumination and extra-abdominal views are more likely to occur during instrument insertion in the preparation phase. The example hard frames can be found in the supplementary material.

To verify the effectiveness of our method, a simple experiment is conducted. The ResNet50 network is respectively trained by the original training set and

Table 1. The statistics of hard frames in the traning set.

	M2CAI16	**Cholec80**
TrocarPlacement	0.27	-
Preparation	0.60	0.45
CalotTriangleDissection	0.22	0.16
ClippingCutting	0.33	0.31
GallbladderDissection	0.28	0.17
GallbladderPackaging	0.27	0.38
CleaningCoagulation	0.29	0.38
GallbladderRetraction	0.40	0.41
Overall	**0.30**	**0.23**

Table 2. The performance gain of ResNet50 after removing hard frames.

	M2CAI16 Workflow Challenge		Cholec80	
	Accuracy ↑	Jacc ↑	Accuracy ↑	Jacc ↑
Clean training set	**1.0%** ↑	**3.1%** ↑	**2.4%** ↑	**4.1%** ↑

the clean training set, from which the hard frames we find are removed. The experiment is repeated 3 times, and the average results are reported. Table 2 shows the performance gain after the hard frames are removed. As the result shows, the existence of hard frames will cause negative impacts to the training process, and it is feasible to mine these hard frames out via data cleansing strategy.

Table 3. Phase recognition results

	M2CAI16		Cholec80	
	Accuracy	Jacc	Accuracy	Jacc
ResNet50	76.3 ± 8.9	56.4 ± 10.4	78.3 ± 7.7	52.2 ± 15.0
PhaseNet [17]	79.5 ± 12.1	64.1 ± 10.3	78.8 ± 4.7	-
EndoNet [16]	-	-	81.7 ± 4.2	-
EndoNet-GTbin [16]	-	-	81.9 ± 4.4	-
SV-RCNet w/o PKI [7]	81.7 ± 8.1	65.4 ± 8.9	85.3 ± 7.3	
Ours[a]	$\mathbf{85.2 \pm 7.5}$	$\mathbf{68.8 \pm 10.5}$	$\mathbf{87.3 \pm 5.7}$	$\mathbf{67.0 \pm 13.3}$
Cadene et al.(nearest online)[4]	86.9 ± 11.0	71.9 ± 12.7	-	-
SV-RCNet + PKI [7]	90.7 ± 6.9	78.2 ± 11.0	92.4 ± 6.9	
Ours + PKI*	$\mathbf{91.2 \pm 5.0}$	$\mathbf{78.7 \pm 13.1}$	$\mathbf{92.4 \pm 5.6}$	$\mathbf{77.0 \pm 11.8}$

[a] We evaluate the experiment on the complete test set for 3 times, and the average results are reported.

4.3 Phase Recognition Results

Table 3 shows a comparison of our solution and others. We first compare our results with the top methods that took part in M2CAI16 Challenge without post-processing strategy. Our solution achieves better performance than the state-of-the-art SV-RCNet [7] by a significant margin, improving accuracy from 81.7% to 85.2% on M2CAI16 Workflow Challenge, and from 85.3% to 87.3% on Cholec80. Note that some methods employ a post-processing scheme for further improvement. To make a fair comparison, we simply modify the PKI [7] post-processing scheme for our solution, denoted as PKI*, which integrates the phase-transition priors. Our final results that integrate post-processing scheme (PKI*) outperform all other approaches.

4.4 Discussion

The main difference between OHFM and PKI scheme is that PKI explicitly use the human-predefined phase transition priors while OHFM learns it from the training data. Compared to PKI scheme and its variants, the OHFM is more general to the unknown surgical videos in real word scenarios and is not sensitive to the hyper parameters. However, it is hard to learn the phase transition well with small amount of training videos.

5 Conclusion and Future Work

In this paper, we focus on the problems of hard frames in surgical phase recognition task and propose a novel solution by detecting hard frames during training and testing. Different from previous works that classify all frames indiscriminately, we first classify those simple frames, and the remaining hard frames are tackled by a further rectifying mechanism. The current results are promising, we believe that there is more room for further improvement in this direction.

Acknowledgement. This work was partially supported by the National Basic Research Program of China (973 Program) under contract 2015CB351803, the Natural Science Foundation of China under contracts 61572042 and 61527804. We also acknowledge the Clinical Medicine Plus X-Young Scholars Project, and High-Performance Computing Platform of Peking University for providing computational resources.

References

1. Beenish, B., Tim, O., Yan, X., Peter, H.: Real-time identification of operating room state from video. In: Proceedings of the 19th Conference on Innovative Applications of Artificial Intelligence, vol. 2, pp. 1761–1766 (2007)
2. Blum, T., Feußner, H., Navab, N.: Modeling and segmentation of surgical workflow from laparoscopic video. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010. LNCS, vol. 6363, pp. 400–407. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15711-0_50

3. Bricon-Souf, N., Newman, C.R.: Context awareness in health care: a review. Int. J. Med. Inform. **76**(1), 2–12 (2007)
4. Cadène, R., Robert, T., Thome, N., Cord, M.: MICCAI workflow challenge: convolutional neural networks with time smoothing and Hidden Markov Model for video frames classification. arxiv abs/1610.05541 (2016)
5. Frenay, B., Verleysen, M.: Classification in the presence of label noise: a survey. IEEE Trans. Neural Netw. Learn. Syst. **25**(5), 845–869 (2014)
6. Gamberger, D., Lavrac, N., Dzeroski, S.: Noise detection and elimination in data preprocessing: experiments in medical domains. Appl. Artif. Intell. **14**(2), 205–223 (2000)
7. Jin, Y., et al.: SV-RCNet: workflow recognition from surgical videos using recurrent convolutional network. IEEE Trans. Med. Imaging **37**(5), 1114–1126 (2018)
8. Lalys, F., Riffaud, L., Bouget, D., Jannin, P.: A framework for the recognition of high-level surgical tasks from video images for cataract surgeries. IEEE Trans. Biomed. Eng. **59**(4), 966–976 (2012)
9. Lalys, F., Riffaud, L., Morandi, X., Jannin, P.: Surgical phases detection from microscope videos by combining SVM and HMM. In: Menze, B., Langs, G., Tu, Z., Criminisi, A. (eds.) MCV 2010. LNCS, vol. 6533, pp. 54–62. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18421-5_6
10. Lin, H.C., Shafran, I., Murphy, T.E., Okamura, A.M., Yuh, D.D., Hager, G.D.: Automatic detection and segmentation of robot-assisted surgical motions. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3749, pp. 802–810. Springer, Heidelberg (2005). https://doi.org/10.1007/11566465_99
11. Miranda, A.L.B., Garcia, L.P.F., Carvalho, A.C.P.L.F., Lorena, A.C.: Use of classification algorithms in noise detection and elimination. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS (LNAI), vol. 5572, pp. 417–424. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02319-4_50
12. Padoy, N., Blum, T., Feussner, H., Berger, M.O., Navab, N.: On-line recognition of surgical activity for monitoring in the operating room. In: Proceedings of the 20th Conference on Innovative Applications of Artificial Intelligence, vol. 3, pp. 1718–1724 (2008)
13. Padoy, N., Blum, T., Ahmadi, S.A., Feussner, H., Berger, M.O., Navab, N.: Statistical modeling and recognition of surgical workflow. Med. Image Anal. **16**(3), 632–641 (2012)
14. Sun, J., Zhao, F., Wang, C., Chen, S.: Identifying and correcting mislabeled training instances. In: Future Generation Communication and Networking (FGCN 2007), vol. 1, pp. 244–250 (2007)
15. Tao, L., Zappella, L., Hager, G.D., Vidal, R.: Surgical gesture segmentation and recognition. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8151, pp. 339–346. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40760-4_43
16. Twinanda, A.P., Shehata, S., et al.: EndoNet: a deep architecture for recognition tasks on laparoscopic videos. IEEE Trans. Med. Imaging **36**(1), 86–97 (2017)
17. Twinanda, A.P., Mutter, D., et al.: Single- and multi-task architectures for surgical workflow challenge at M2CAI 2016. arxiv abs/1610.08844 (2016)