SV-RCNet: Workflow Recognition From Surgical Videos Using Recurrent Convolutional Network

Chengbo Zang (cz2678)
cz2678@columbia.edu

Yuqing Cao (yc3998)
yc3998@columbia.edu

Fengyang Shang (fs2752)
fs2752@columbia.edu

Abstract—This project reproduced the basic structure and worflow of SV-RCNet for surgical phase prediction proposed by the original paper. Necessary modifications were made with respect to type of baseline feature extractors and different hyperparameters. Multiple regularization techniques including augmentation, dropout and weight-decay were introduced and the effectiveness was studied under experiments. The ResNet+LSTM structure is proved to be competent in learning and predicting surgical phases by respectively capturing visual features and temporal information. The model achieved 78.63% accuracy on Kaggle test sets.

Keywords—SV-RCNet, Surgical phase recognition, ResNet, LSTM

I. Model Design

A. SV-RCNet Architecture

Since the hernia surgical video dataset is a typical type of sequential data, it is crucial to leverage the temporal information of the videos, apart from effort in extracting the features of each frame, in order to perform well in the phase classification task. Additionally, it has been addressed in [1] that the model sensitivity to subtle or non-linear sequential dynamics plays a significant role in exploiting temporal information. Therefore, it is challenging to only use shallow CNN based neural networks [2] to solve this issue. In [1], they proposed a model named SV-RCNet utilizing both ResNet [2] and LSTM [3], as shown in Fig.1. The deep ResNet makes the model powerful to extract high-level visual features and the LSTM part contributes to processing sequential data in a non-linear way of modeling long-range temporal dependencies. They claimed that one advantage of SV-RCNet is to integrate these two parts and train them end-to-end jointly, in contrast to most exiting models that separately harness spatial and temporal information [4-9].

To be more specific, they demonstrated a set of residual blocks in Fig.2 (b). They used x_l and x_{l+1} to denote the input and output features accordingly, with regard to the *l-th* block. Eq. (1) shows the residual mapping for them, where W_s is a linear matrix that matches the input and the output dimensions, F_l is the residual mapping function, and W_l represents the weights involved in the *l-th* block.

$$x_{l+1} = W_s x_l + F_1(x_l; W_l) \tag{1}$$

They also shew that there are three convolutional layers followed by one batch normalization layer and one ReLU layer in the ResBlock. And at the beginning of the ResNet, the input images would go through a 7 × 7 convolutional layer and a 3 × 3 max pooling later for down sampling.

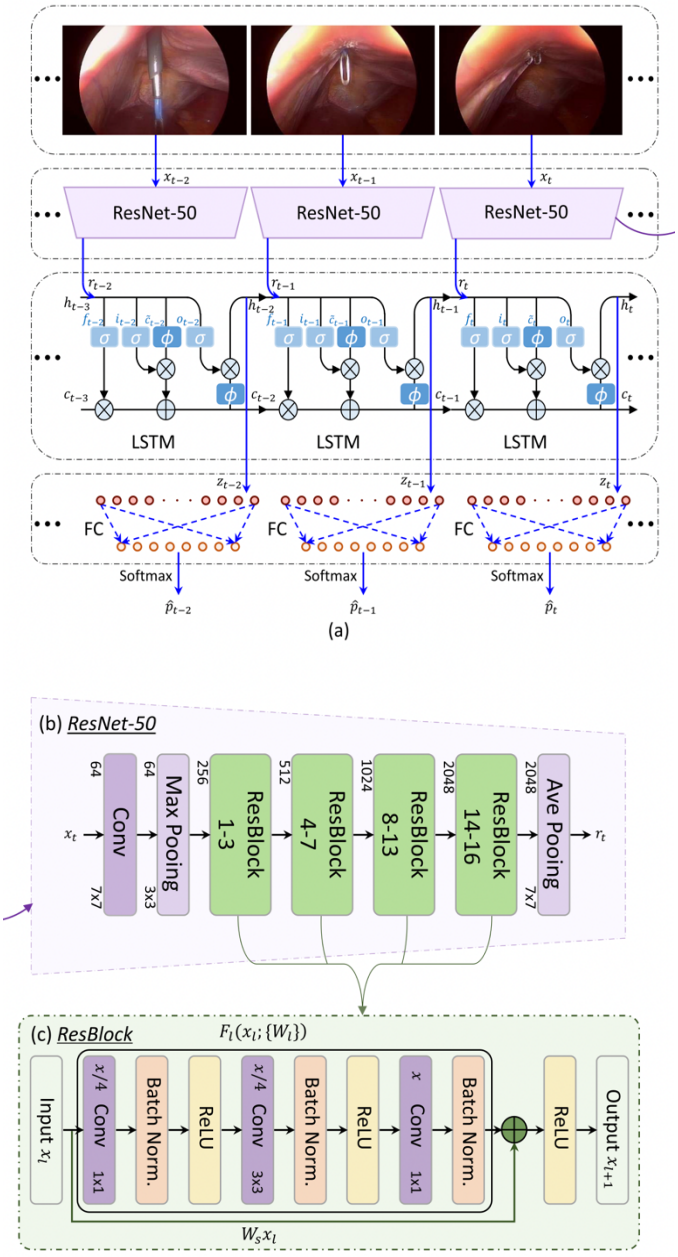

Fig. 1. The SV-RCNet model proposed in [1]. (a) An overview of the model. (b) The architecture of ResNet for extracting visual features from each frame. (c) The ResBlock implemented in the ResNet.

Although in [1] they only chose ResNet-50 for the model, in our project, we will implement ResNet-18 and ResNet-50 in our SV-RCNet models and respectively conduct experiments to compare their discrimination performance. Hopefully in this way, we could get more insight into the importance of the depth of ResNet and its influence at the generalization ability of the entire model.

B. *End-to-End Learning*

They emphasized that the SV-RCNet is trained in an End-to-End manner, so as to best exploit the complementary information. The parameters for ResNet and LSTM are optimized jointly, instead of learning visual and temporal features independently. The forward process of ResNet is denoted by R_β, where β represents its weights. For each input frame x_j, the ResNet serves as a visual descriptor extraction, expressed as $r_j = R\beta(x_j)$. Subsequently, these output features $r = \{r_{t'}, \dots, r_{t-1}, r_t\}$ would been fed into the LSTM part sequentially, denoted by L_θ, where θ represents its parameters. Given input features r_t and previous hidden state h_t, the forwarding output of LSTM is $z_t = h_t = L_\theta(r_t, h_{t-1})$. Eventually, our SV-RCNet output is $O_t = W_z z_t + b_z$, where W_z and b_z are weights and bias accordingly, and O_t is a fourteen-dimensional vector corresponding to each phase logit at time t.

Let O_t^c to represent the *c-th* element of the vector O_t, then $O_t^c = 1$ if class c is the inferred phase for the frame at time t, or otherwise $O_t^c = 0$. Given the ground truth label l_t of that frame, the overall loss function for the SV-RCNet model could be formulated as Eq. (2):

$$L(X;\beta,\theta) = \frac{1}{N}\sum_{x\in X} l(\mathrm{x})$$
$$= -\frac{1}{N}\sum_{x\in X}\ \sum_{\tau=t'}^{t} log O_\tau^{l_\tau}(x_{t':\tau}, h_{t':\tau};\beta,\theta) \quad (2)$$

where X represents the training dataset with length of *N*. It could be seen that in this back propagation procedure, the parameters for ResNet β and those for LSTM θ are jointly optimized. In our project we implement *Adam* as the gradient descent method. In [1], they also proved that the optimization equation for β involves θ. As a result, updates of θ will influence the learning process of β and vice versa.

II. METHODOLOGIES

Following the workflow purposed by the original paper, we process the surgery videos frame by frame, using classical CNN as the feature extractor, along with LSTM to incorporate the temporal information.

A. *Preprocessing*

The data preprocessing involved in this project mainly refers to extracting frames from the videos and augment them.

1) Frame Extraction: All the frames in the videos are first extracted and converted to images with FFmpeg [10] command line tool. Although this procedure takes up sigificantly more amount of storage, the data loading and label assignment are relatively straightforward.

Being a classification problem in nature, the model has to learn to recognize the genreal charateristics of a particualr phase across different videos, so the labels of the same phase in all videos are required to be uniform. To deal with the initialy noisy labels (mainly inconsistent letter cases and spelling errors), we started by collecting all labels (both correct and incorrect) and manually selected out the true ones to form a template. When the noisy labels are trying to be assigned to each frame, a string similarity check [11] is performed between the suscepible label and the template. We followed non maximum suppression to determine the final ground truth label for each frame. This issue is fixed therefore depricated in later versions of training data provided.

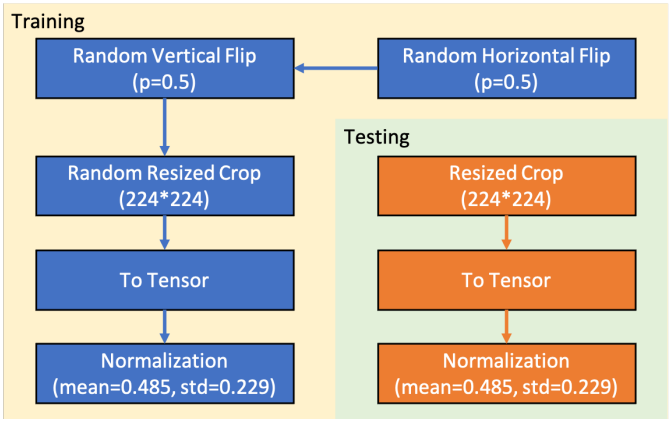

Fig. 2. Data preprocessing and augmentation for training (left) and testing (right) phase respectively.

2) Augmentation: To increase the generalizability of the model on the test set and alleviate overfitting (both the resnet and LSTM are large in capacity), some widely used augmentation techniques are applied to the data before they are fed into the network [12]. As is shown in Fig. x, in training phase, each image are flipped both horizontally and vertically with a probablility of 0.5, and resized to 224 × 224 with original scale randomly cropped within range of (0.4 - 1.0). Then, the pixel values are converted to tensor before normalized to have mean of 0.485 and standard deviation of 0.229. There's no random flipping or cropping in validation and testing, while resizing and normalization are performed in the same manner.

Table 1. ResNet-18 Structure

Layer Num	Output Size	ResNet-18
conv1	64*112*112	64, 7*7, stride 2
conv2_x	64*56*56	3*3 max pool
		(64, 3*3) * 2 * 2
conv3_x	128*28*28	(128, 3*3) *2 * 2
conv4_x	256*14*14	(256, 3*3) *2 * 2
conv5_x	512*7*7	(512, 3*3) *2 * 2
average pool	512*1*1	7*7 average pool

B. *Feature Extraction*

The model utilize a CNN as baseline feature extractor to charaterize individual frames with no regard to sequential information. While ResNet50 was adopted by the original paper, we experimented with several simpler strcutures

including ResNet18 pretrained on ImageNet [13] and a simple 3-layer CNN considering the amount of data and difficulty of training. The structure of Resnet18 is given in Table 1.

C. Regularization

1) Weight Decay: It is observed during the training that both the ResNet and LSTM are prone to overfitting (see section III), making regularization an important consideration in pipline design. Here we utilized weight decay of Adam optimizer, of which the update rule is given by

$$w \leftarrow (1-\lambda)w - \eta \nabla L_0 \tag{3}$$

where λ is the decay factor, η is the learning rate and L_0 denotes the unregularized loss function.

This is equivalent to performing an L2 regularization that is properly parametrized [14]. Specifically, a typical loss function with L2 regularization term is given by

$$L = L_0 + \frac{\lambda'}{2}\|w\|_2^2 \tag{4}$$

where λ' is the penalty coefficient. Then the weight can be updated as

$$\begin{aligned} w &\leftarrow w - \eta \nabla L \\ w &\leftarrow w - \eta \nabla L_0 - \eta \lambda w \\ w &\leftarrow (1-\eta\lambda)w - \eta \nabla L_0 \end{aligned} \tag{5}$$

Let $\lambda = \lambda'$ and we are obtained with Eq. 3 for standard weight decay. This is expected to gap the difference between the modeling of training and testing sets, and the experimental results are also described in section III.

2) Dropout: Dropout are introduced both to the fully connected layer after ResNet18 and the LSTM, both set to a probability of 0.5 [15].

D. Time Temporal Information

Although experiments ahve shown that ResNet along is working effectively to some extent, it is obvious that surgical phases have strong sequential information in nature, and the model performances are expected to improve if this is succesfully incorperated. Therefore, we followed the original paper to append an LSTM layer. A stateful archetecture [15] was used where the latent states computed by the last time step are stored and used as inputs for the next time step. Necessary adjustments on batch size (number of time sequneces for LSTM) were made with consideration of the difference in frame rates between our data and the original paper, as well as the limitation of computational power and training time.

III. Experiments & Results

A. Dataset

We used SV-RCNet on the dataset supported by the class. The dataset is consisted of 70 training and more testing cideos videos recording the hernia reduction. These videos are acquired at 1 frame per second and the resolution of each frame is 854 × 480. And all these images can be catagorized into 14 phases. We divided such labeled videos into the training set (50 videos) and the validation set (20 videos).

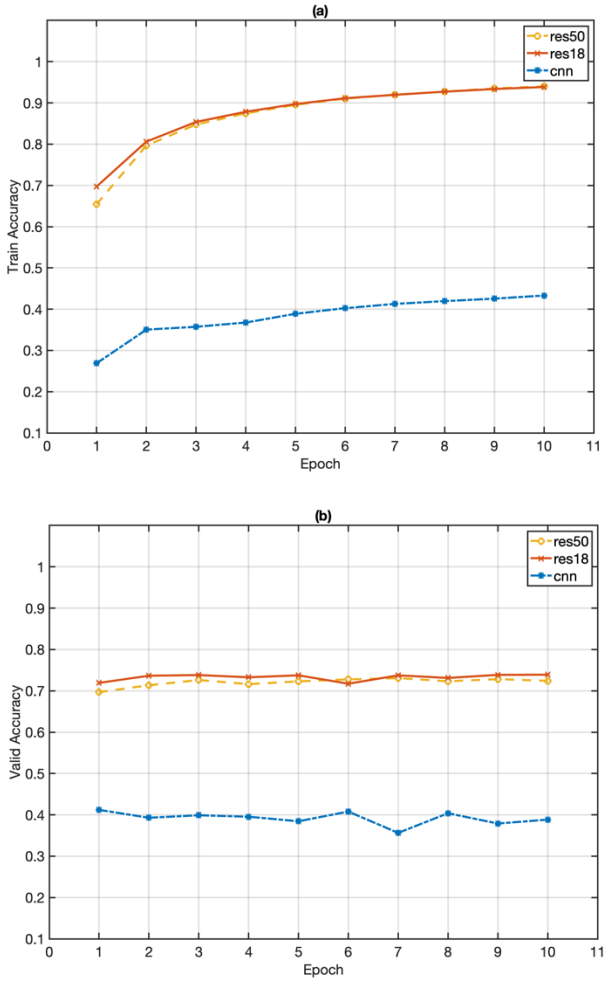

Fig. 3. (a) Training accuracy of different baselines (Simpe CNN, ResNet-18 and ResNet-50). (b) Validation accuracy of different baselines (Simpe CNN, ResNet-18 and ResNet-50).

B. Experiments

1) Baselines: Extracting visual features from the frames is the most crucial part to identify the surgery phases. Choosing the best network model will benefit both time complexity and accuracy. So, we implemented three convolutional networks (3 layers CNN, ResNet-18 and ResNet-50) to find out which is the best model with higher accuracy and lower time consumption for this task.

Note that in our tasks, the performances of the model are obtained directly from the accuracy of both training sets and validation sets. From Fig. 3, it can be seen that compared with CNN, both ResNet-18 and ResNet-50 perform very well on this task with high training and validation accuracy. Also, the depth of the network is the main factor that influenced the time complexity. With nearly the same accuracy for this task, we decided to choose ResNet-18 with fewer layers to be the model extracting the visual features from the images, while preserving large enough capacity.

On top of ResNet-18, we tried to further enhance the performance of predicting the surgeon's phases. Considering the dataset we used is not independent images, they are consecutive frames extracted from the videos. As a matter of

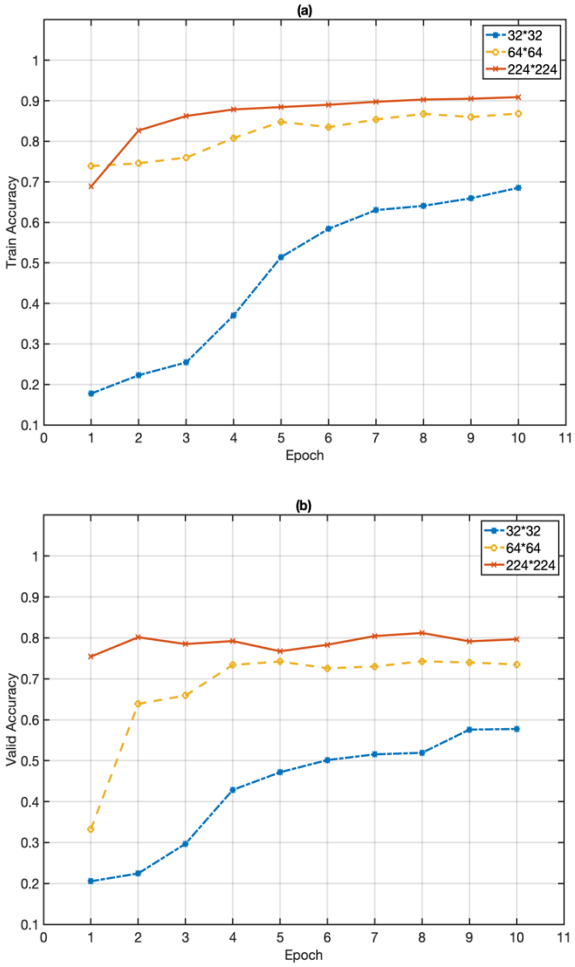

Fig. 4. (a) Training accuracy of different input sizes (32, 64 and 224). (b) Validation accuracy of different input sizes (32, 64 and 224).

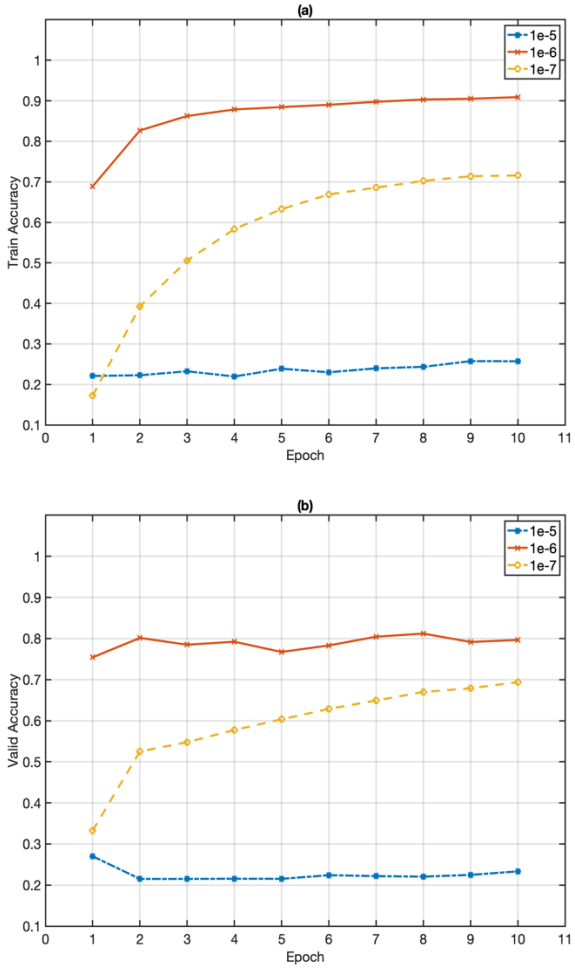

Fig. 5. (a) Training accuracy of different learning rates (1e-5, 5e-6 and 1e-7). (b) Validation accuracy of different learning rates (1e-5, 5e-6 and 1e-7).

fact, we need to leverage the ordering information from the videos to enhance the performance of identifying phases. For such reasons, ResNet-18 followed by LSTM is chosen as the model for our experiments, denoted by the original paper as SV-RCNet.

2) Input sizes: We used different input sizes to check how much it will affect our final result with the same model and the same parameters.

In Fig. 4, three different lines indicate different input sizes as is shown in the legend. Theoretically, the more complex the model, the more information is used to make a good prediction. Our model has considerable complexity. Hence, it can be intuitively observed from Fig. 4 that the higher input sizes are, the higher accuracy is our predictions.

3) Learning rates: We then further experimented with some other parameters that may affect the model performance. The first is the most important hyperparameter for the model, the learning rate. We really want to know what is the range of the learning rate that may give us better trade-off between the time we spent and the accuracy we can get.

In this part, we chose three different learning rates for model trainig (1e-5, 5e-6 and 1e-7). With the same epochs, the time consumption for the training of each model can be compared.

And for performances, we can directly compare the accuracy. From the Fig. 5, we can find using 1e-5 is too high to train the model and may results in difficulty in convergence to the global minimum. Although using 1e-7 seems great, and it can train to get better performance. But it will be more time consuming than using a little bit higher learning rate. So in this experiment, we find our best learning rate is 5e-6.

Then from the experiments above, we found that the training accuracy is usually higher than the validation accuracy. In this regard, this reminds us that our model is a little bit overfitting. Theoretically, this can be solved by regularization. Hence, we utilize two ways to regularize our model, dropout layers and L2 norm.

However, it can be observed from Fig. 6 that although both of them seems to be effective in traing phase, they did little to alleviate overfitting during validations. Specifically, comparing blue star line and the red line, dropout layers counterintuitively made the performance worse than usual. But using both weight decay in the loss function and dropout layers help the model get better performances than three other models.

In a word, the utilization of regularization is helpful for solving the overfitting in our model to some extent. We chose to keep theses features during the training of our model.

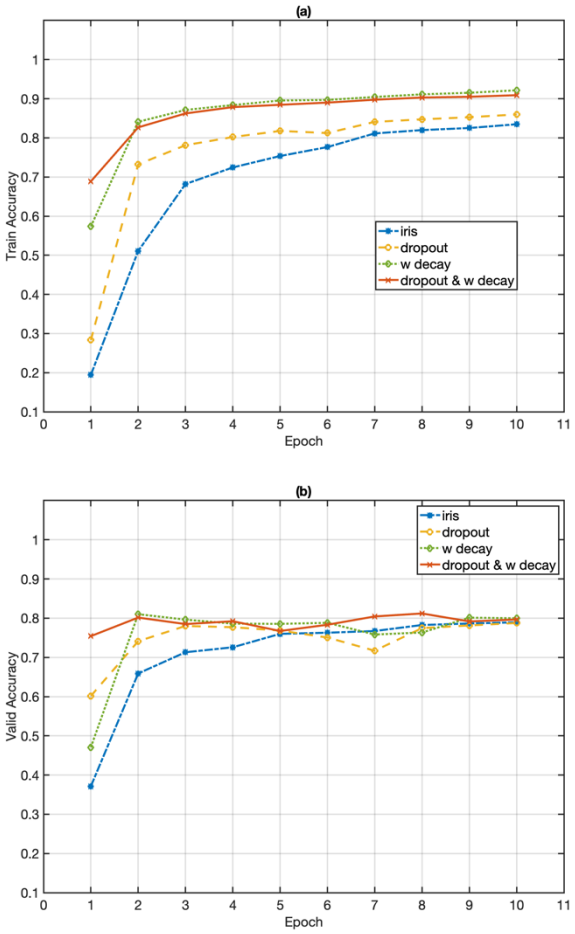

Fig. 6. (a) Training accuracy with different regularization techniques (iris model, dropout, weight decay and both). (b) Validation accuracy with different regularization techniques (iris model, dropout, weight decay and both).

IV. Discussion & Conclusion

In this project, we reproduced the basic model structure and workflow for surgical phase prediction using SV-RCNet. Necessary modifications were made with regard to baseline feature extractors and training techniques. It can be observed from Table 2 that SV-RCNet gives better performance than all other models (simple CNN, ResNet-18 and ResNet-50) in validation accuracy, macro-F1 score and AUC. Class-wise precision-recall is then evaluated on "hernia reduction", which is assumed to be the most important phase to be segmented. SV-RCNet still remained to be the best model across comparisons.

Indeed, several aspects were concluded from the experiments that can be further studied and improved.

First of all, we found in our experiments that the requirement of the complexity for baseline feature extractors increase significantly with the growing amount of training data (both the input image size and the number of videos). Although ResNet-18 was shown to have enough capability to extract meaningful features, more complex structures like ResNet-50 are expected to yield better performance under sufficient training.

Secondly, in our expriments the temporal information embedded in the videos are expected to be modeled by LSTM which is trained from scratch. This fails to utilize prior knowledge about the surgical procedure that can be easily deduced by human while hard to learn by machine. A Prior Knowledge Inference (PKI) algorithm was proposed by the original paper to address this issue, but the effectiveness of this algorithm as well as other sequential modeling techneques in our scenario is still under investigation.

Thirdly, some popular tricks (mainly dropout and weight-decay) battling overfitting present limited effects according to our experiments. This may be resulted by inappropriate regularization parameters considering the insufficient training conducted. Additionally, the influence of other techniques like L1 regularization have not been studied.

Last but not least, trade-offs had to be made between the hands-on computational rescources and a more reasonable training process. Fine tuning of hyperparameters of the model and regularization process were also restricted.

Table 2. Model Comparisons
(class-wise precison-recall evaluated on "hernia reduction" phase)

Model	Acc.	Prec.	Rec.	M. F1	AUC
CNN	0.665	0.451	0.295	0.305	0.819
ResNet-18	0.822	0.704	0.633	0.639	0.846
ResNet-50	0.731	0.549	0.575	0.548	0.834
SVRC	**0.887**	**0.870**	**0.778**	**0.806**	**0.854**

References

[1] Jin, Y., Dou, Q., Chen, H., Yu, L., Qin, J., Fu, C. W., & Heng, P. A. (2017). SV-RCNet: workflow recognition from surgical videos using recurrent convolutional network. *IEEE transactions on medical imaging*, *37*(5), 1114-1126.

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[4] 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.*, vol. 36, no. 1, pp. 86–97, Jan. 2017.

[5] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1153–1159, May 2016.

[6] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annu. Rev. Biomed. Eng.*, vol. 19, pp. 221–248, Jun. 2017.

[7] L. Tao, L. Zappella, G. D. Hager, and R. Vidal, "Surgical gesture segmentation and recognition," in *Proc. Int. Conf. Med. Image Comput.-Assist. Intervent.*, 2013, pp. 339–346.

[8] F. Lalys, L. Riffaud, X. Morandi, and P. Jannin, "Surgical phases detection from microscope videos by combining SVM and HMM," in *Proc. Int. MICCAI Workshop Med. Comput. Vis.*, 2010, pp. 54–62.

[9] F. Lalys, L. Riffaud, D. Bouget, and P. Jannin, "A framework for the recognition of high-level surgical tasks from video images for cataract surgeries," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 4, pp. 966–976, Apr. 2012.

[10] Tomar, S. (2006). Converting video formats with FFmpeg. Linux Journal, 2006(146), 10.

[11] Ratcliff, J. W., & Metzener, D. E. (1988). Pattern-matching-the gestalt approach. Dr Dobbs Journal, 13(7), 46.

[12] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of big data, 6(1), 1-48.

[13] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[14] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

[15] Marafioti, A., Hayoz, M., Gallardo, M., Márquez Neila, P., Wolf, S., Zinkernagel, M., & Sznitman, R. (2021, September). CataNet: Predicting remaining cataract surgery duration. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 426-435). Springer, Cham.

MEMBERS

Name	UNI	Task
Yuqing Cao	yc3998	Model Construction Fine tuning
Fengyang Shang	fs2752	Model training Fine tuning
Chengbo Zang	cz2678	Data processing Fine tuning