Surgical Workflow Anticipation

By

Kun Yuan

A thesis submitted in partial fulfillment of the requirements for the
Master's degree in Computer Science Concentration in Applied Artificial Intelligence

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

University of Ottawa

October 2021

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. The reprint permission is obtained through the publishers.

ABSTRACT

As a non-robotic minimally invasive surgery, endoscopic surgery is one of the widely used surgeries for the medical domain to reduce the risk of infection, incisions, and the discomfort of the patient. The endoscopic surgery procedure, also named surgical workflow in this work, can be divided into different sub-phases. During the procedure, the surgeon inserts a thin, flexible tube with a video camera through a small incision or a natural orifice like the mouth or nostrils. The surgeon can utilize tiny surgical instruments while viewing organs on the computer monitor through these tubes. The surgery only allows a limited number of instruments simultaneously appearing in the body, requiring a sufficient instrument preparation method. Therefore, surgical workflow anticipation, including surgical instrument and phase anticipation, is essential for an intra-operative decision-support system. It deciphers the surgeon's behaviors and the patient's status to forecast surgical instrument and phase occurrence before they appear, supporting instrument preparation and computer-assisted intervention (CAI) systems. In this work, we investigate an unexplored surgical workflow anticipation problem by proposing an Instrument Interaction Aware Anticipation Network (IIA-Net). Spatially, it utilizes rich visual features about the context information around the instrument, i.e., instrument interaction with their surroundings. Temporally, it allows for a large receptive field to capture the long-term dependency in the long and untrimmed surgical videos through a causal dilated multi-stage temporal convolutional network. Our model enforces an online inference with reliable predictions even with severe noise and artifacts in the recorded videos. Extensive experiments on Cholec80 dataset demonstrate the performance of our proposed method exceeds the state-of-the-art method by a large margin (1.40 v.s. 1.75 for inMAE and 2.14 v.s. 2.68 for eMAE).

PUBLICATION DURING MASTER'S

- **Yuan, K.**, Shijian Gao, Matthew Holden Lee, W. (2021, June). Surgical Workflow Anticipation using Instrument Interaction. In the 24th International Conference on Medical Image Computing and Computer Assisted Intervention, MICCAI 2021.
- Ratul, M. A. R., **Yuan, K.**, Lee, W. (2021, April). CCX-rayNet: A Class Conditioned Convolutional Neural Network For Biplanar X-Rays to CT Volume. In 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI) (pp. 1655-1659). IEEE.
- Ratul, M. A. R., Elahi, M. T., **Yuan, K.**, Lee, W. (2020, December). RAM-Net: A Residual Attention MobileNet to Detect COVID-19 Cases from Chest X-Ray Images. In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 195-200). IEEE.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisors Professor WonSook Lee and Professor Matthew Holden for giving me unlimited support and inspiration over the past two years. They always helps me to think intuitively to pursue any research questions and encourages me to keep on track to achieve my dream. I can not be able to accomplish this degree without their proper guidance and dedicated involvement in every step of my research work. In the last two years, I have come and contact with some incredible and helpful lab members. They assist me throughout my time at the University of Ottawa with constructive feedback, support, and in-depth knowledge. Last but not least, I would like to acknowledge the tremendous love, motivation, and immense support my family and friends had given me. Specifically, without the inspiration from my parents, I would not be able to reach here.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The surgical practice has significantly evolved throughout the centuries. It underwent revolutionary changes with the introduction of anesthesia and antiseptics in the 19th century. Surgery is a profession defined by its authority to cure by means of bodily invasion [23] . It keeps upgrading the technique on the diagnosis and treatment of various medical conditions, expecting a safe outcome for the patients from the surgery. This is crucial because the studies show that 9 million of and 300 million surgical procedures per year worldwide will encounter major complications [81, 82]. Also, how well the surgery is performed is highly relevant to the readmission rate and can be a good reference for the hospital to optimize the operating room (OR).

In the 20th century, advances in surgery had been made centered around professionalization, systematic measurement of outcomes of care, and minimally invasive access to surgical sites [52]. To achieve these objectives, multiple techniques, such as multi-modal medical imaging technology [7], the development of surgical microscopes and endoscopes has made a dominant influence on the upgrading process. Also, the computer-assisted interventional surgery [9] has been improved thanks to the recent emergence of deep learning methods. Furthermore, the internet revolution has brought access to an almost unlimited amount of electronic patient records and can be another modality data for surgical data science. However, this text-based data is not our focus and has no direct integration with the computer-assisted surgical system. It needs the seamless integration of computer-aids in the surgical environment, which enhances situation awareness, ergonomics, and minimization of cognitive workload [52], and it has not yet been achieved.

Towards the safe, effective and efficient surgical process, this work focuses on gallbladder removal surgery and investigates an anticipation problem. This work mainly focuses on providing real-time auxiliary signals to help the surgeon's decision-making and link decisions to patient outcomes. This will enable high-quality treatment and will ensure an efficient OR

optimization process. For the patient, this allows them access to the best surgical care from both human experts and machine experts. The surgery will only be influenced by the variability arising from unique patient characteristics rather than the choice of surgeon or care facility. Also, our work can be deployed into the surgical coaching process to improve the surgeon's skills and assess them. Benefited from the recent emerging deep learning methods, our work becomes plausible and makes surgical data science forward to create "superhuman" surgery. The work is evaluated on the public large dataset extensively.

1.1 Motivation

While surgical data science is related to the field of biomedical data science, its unique characteristic is the focus on procedural data. It pertains to (i) the patient, (ii) effectors involved in the manipulation of the patient including physicians, anesthesia team, nurses and devices, including robots, (iii) sensors for perceiving patient- and procedure-related data such as images, vital signs, medical device data and motion data as well as (iv) domain knowledge, including factual knowledge, such as (hospital-specific) standards related to the clinical workflow, previous findings from studies or clinical guidelines as well as practical knowledge from previous procedures.

Context-aware assistance is integral for CAI systems, of which the most crucial task is surgical workflow anticipation. It anticipates the occurrence of surgical instruments and phases before they appear, enabling the efficient instrument preparation and intelligent robot assistance system design [18, 63]. Also, it can increase patient safety, reduce surgical errors and optimize communication in the operating room (OR) [52]. For example, anticipating the surgical instrument's usage can provide vital input to physicians in the form of early warning in cases of deviations and anomalies. Anticipating the surgical phases can also help a robotic system identify events, such as bleeding, beforehand and decide when to intervene.

Recent models [63,75] for surgical workflow anticipation possess spatial-temporal limitations. Spatially, they use AlexNet [43], VGG [70] and similar architectures to extract a feature vector, representing instrument/phase presence for each frame. However, they ignore the task-specific combinations present in surgical anticipation applications, i.e., instrument-

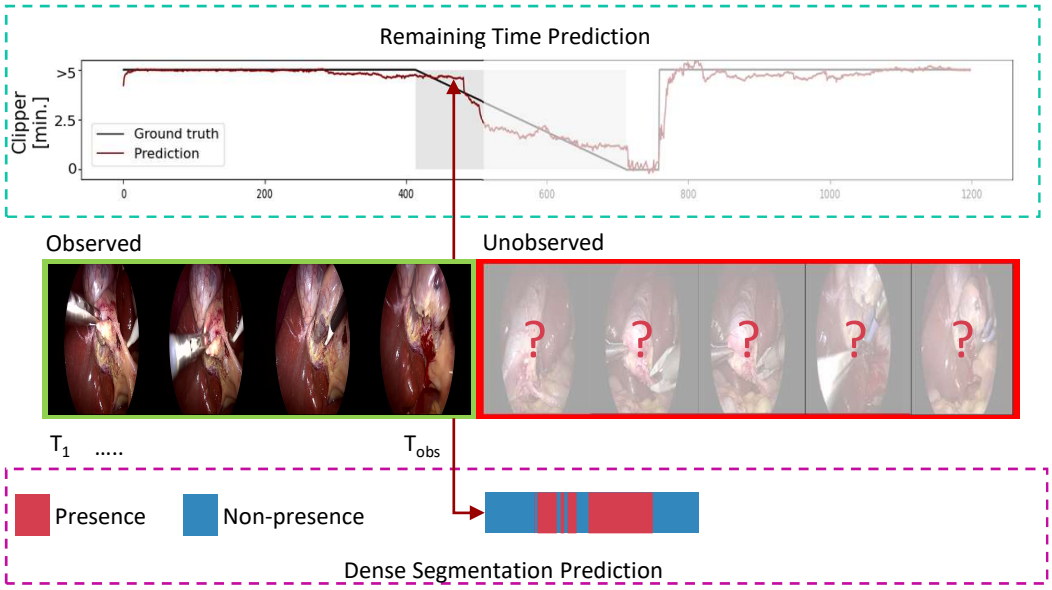


Figure 1.1: Anticipation frameworks. Given an observed sequence and current time instant, T_{obs}, bottom part shows the conventional anticipation works that predicts dense segmentations. It outputs a 0/1 sequence to indicate the future's non-presence/presence. The upper part shows our strategy handling anticipation task as a real-time remaining time prediction task. It outputs the remaining time until the next occurrence.

instrument and instrument-surrounding interactions. This information precisely reflects the surgeon's intention and patient's anatomy status, helping models generalize to the low-quality input materials [42] and variability of patient's anatomy and surgeon style [20]. Our IIA-Net addresses instrument-instrument interaction in the form of a correlation matrix and designed geometric relations among instruments. Also, the instrument-surrounding interaction is included via the semantic segmentation map. This makes our extracted feature to be representative enough to identify the trigger event for the next instrument and phase occurrence.

Temporally, existing works have difficulty handling non-stationary time series. Especially for surgical workflow, e.g., laparoscopic surgery, whose transitions among instruments and phases are ambiguous and various. This requires the temporal modeling method to integrate recent observations with the long-range context in a computationally efficient way. However, widely used RNNs [30] learn a pattern from shorts segments of time series and apply it to other parts to get predictions, losing the distant observation information. Therefore, we

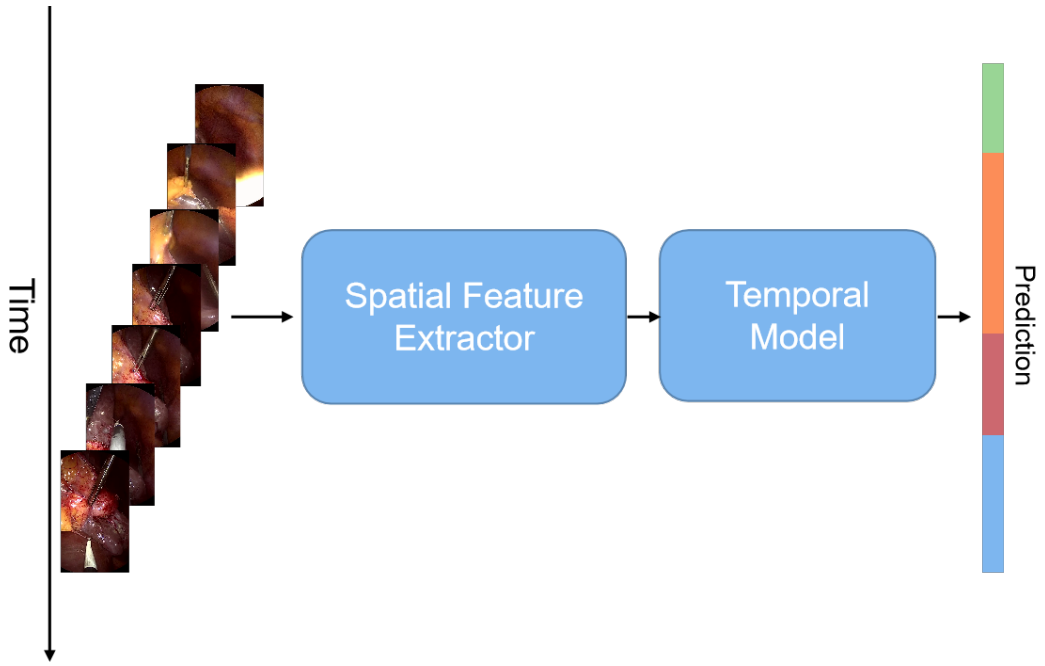


Figure 1.2: Overview of the system. It consists of two step of trainable networks, spatial feature extractor and temporal model. We propose the IIM module into the spatial feature extractor and MS-TCN into the temporal model.

opt for dilated temporal convolutions to handle the full resolution of time series. This aids temporal pattern modeling and does not require complex computational resources.

Also, initial works [2,11,18,21,39,51] handle anticipation as a dense segmentation prediction task, shown in the bottom of Figure. 1.1. They require a pre-loading process before performing anticipation, limiting their usage in online surgical applications. Specifically, Abu Farha et al. [2] propose an anticipation system that needs to observe at least 10%/20% of the video before it starts the prediction. Figure. 1.1 shows an example where the predicted dense segmentation usually contains the short segments, which are ambiguous to determine the trending of the instrument's presence.

1.2 Overview of the System

In this work, IIA-Net achieves workflow anticipation tasks facilitated by their instrument-instrument and instrument-surrounding interaction prior. Unlike the previous methods, the raw frame is directly sent to the deep learning network to extract visual features and model

temporal dependency in an end-to-end training manner, this work follows the work of [10] to split the whole process into spatial feature extraction step and temporal modeling step, as shown in Figure. 1.2. For the spatial feature extractor, we propose an Instrument Interaction Module (IIM) to extract the relation among instruments. Also, the location information indicating the interaction between surgical instruments and anatomical structures is introduced by the IIM. These priors capture the surgeon's intention through recognizing the action performed by the instrument. However, these clues are beneficial for triggering activity identification but are neglected and not be fused in the former works. For the temporal modeling step, we propose to use the MS-TCN [14] to replace the widely used LSTM due to its large receptive field and low computational cost. This allows the CAI system a real-time inference capability and is applicable to the online surgical scenario.

In particular, we show that the interaction prior is crucial for anticipation of surgical instruments and phases. During the surgery, different surgical phases and instruments start with specific action patterns from the surgeon. Inspired by the work of [47], we model the interaction explicitly instead of implicitly by simply utilizing a deep learning network. In this way, we can capture the surgeon's intention from the instrument movement and the patient's inner body status. To apply the model for fast and accurate prediction, we utilize the causal MS-TCN to model the temporal dependency for each frame. Specifically, each frame outputs a regression result, indicating the remaining time until the occurrence of the next phase or instrument. Unlike the Vanilla MS-TCN that predicts the remaining time based on the former frames and consecutive frames, our designed causal convolution only considers the previous frames to predict the current timestamp. This makes our system feasible for online inference.

1.3 Contribution

Our contribution is four-fold:

- Spatially, we propose a novel instrument interaction module (IIM) for the feature extraction process.
- Temporally, we apply, for the first time, the causal dilated multi-stage temporal convolu-

tional network (MS-TCN) structure to surgical workflow anticipation, with an accurate and fast online inference.

- We combine spatial and temporal information to form a two-step IIA-Net for surgical workflow anticipation.
- We propose a multi-task learning schema to jointly anticipate instrument and phase occurrence, which are important challenges in surgical workflow anticipation.

1.4 Structure of the Thesis

The rest of the thesis is arranged into the following chapters:

- Chapter 2 provides a brief literature review, including the deep learning background, interventional heal care applications, spatial feature extraction, anticipation methods, temporal modeling methods and multi-task learning.
- Chapter 3 presents our methodology by explains the task formulation of the anticipation problem, the design of the network architecture, choice of the loss functions, and the strategy of the multi-task learning
- Chapter 4 shows the experiment setups of this work. It indicates the dataset used for training the model, detailed hyperparameters for the training process and the evaluation metrics for the analysis and discussion.
- Chapter 5 elaborates on the anticipation results, effect of IIM and stages in MS-TCN, sample results of segmentation, sample results of tool detection, correlations, phase transition and limitations of this work.
- Chapter 6 concludes the thesis work. It also outlines the significant contributions, potential applications, and the directions of the future work.

Chapter 2

LITERATURE REVIEW

Deep learning, especially deep convolutional neural networks, have achieved a series of breakthroughs for academia and industry in recent years. Without relying on the hand-crafted features designed by the human expert, the deep convolutional neural network can adaptively extract task-specific features and solve various complex problems thanks to its strong non-linear approximation ability. It naturally integrates different levels of features in an end-to-end fashion, leading the main direction of artificial intelligence development. The basic knowledge needed for understanding this thesis is as follows.

2.1 Deep Learning Background

2.1.1 Convolutional Neural Network (CNN)

Convolutional neural network (CNN) [45], inspired by the human's vision system, extracts feature from the local response. Specifically, it focuses on the combination of one pixel and other adjacent pixels using the fixed-size convolutional kernels. Each kernel is responsible for extracting a channel of feature map from an image, and the feature maps are fed to the multilayer perceptron classifier for many computer vision tasks. Compared with the fully connected neural network, the convolutional neural network greatly reduces the computational cost by decreasing the number of parameters that are required for model training. As the most popular architecture in the field of deep learning , the convolutional neural network (CNN) is widely used in image recognition, object detection, image segmentation, etc. Its derivatives such as VGG [70], ResNet [27], DesneNet [31] are applied in many real-world applications. It consists of four main components, convolutional layer, pooling layer, normalization layer, and dropout layer, to improve the efficiency and effectiveness. In addition, in video understanding tasks, the temporal convolutional network [44] is commonly applied to capture the sequential dependencies and perform temporal modeling.

- Convolutional Layer

The convolutional layer, as the most important component of the CNN, can be classified into three categories, 1D, 2D, and 3D convolutional layer, depending on the input format. The aim of convolution layer is to learn a set of convolution kernels and extract useful patterns and features with those. At the initial stage of the training process, weights of the kernels are randomly initialized, and they will be optimized through back-propagation during the training process. Eventually, the convolution kernel will be effective enough to generate representative features from input data for different computer vision tasks. As shown in Figure. 2.1, the kernel slides on the height and width of the image input and produces the dot product of the kernel and the image at each spatial position. The length of the grain sliding is called the stride length. Convolution operation emphasizes the area of the image that displayed the valuable features. The result of this convolution layer is known as a feature map.

- Pooling Layer

The second module that makes CNN powerful is the pooling strategy, which has been applied to many typical CNN structures VGGsimonyan2014very, AlexNet [43]. Similar to the convolution operation, pooling is a vector that performs a scalar transformation on each local area of the image. However, it has no filters to calculate the dot product with the local area. Instead, it follows the heuristic strategy to shrink the spatial dimension. For example, average pooling will represent a window of pixels using the average of them, max pooling will simply pick the largest pixel and discard the rest.

As shown in the Figure. 2.2, it is 2×2 pooling, which can effectively reduce the size of the feature map by 2 times. The idea of pooling may seem counterproductive because it will cause the loss of information. Still, it has been proven to be very effective in practice because it makes CNN invariant to changes in image representation, and it also reduces the effect of background noise. Max pooling has worked best in recent years because the largest pixel in a local area usually represents the most important feature of the area. Usually, the image

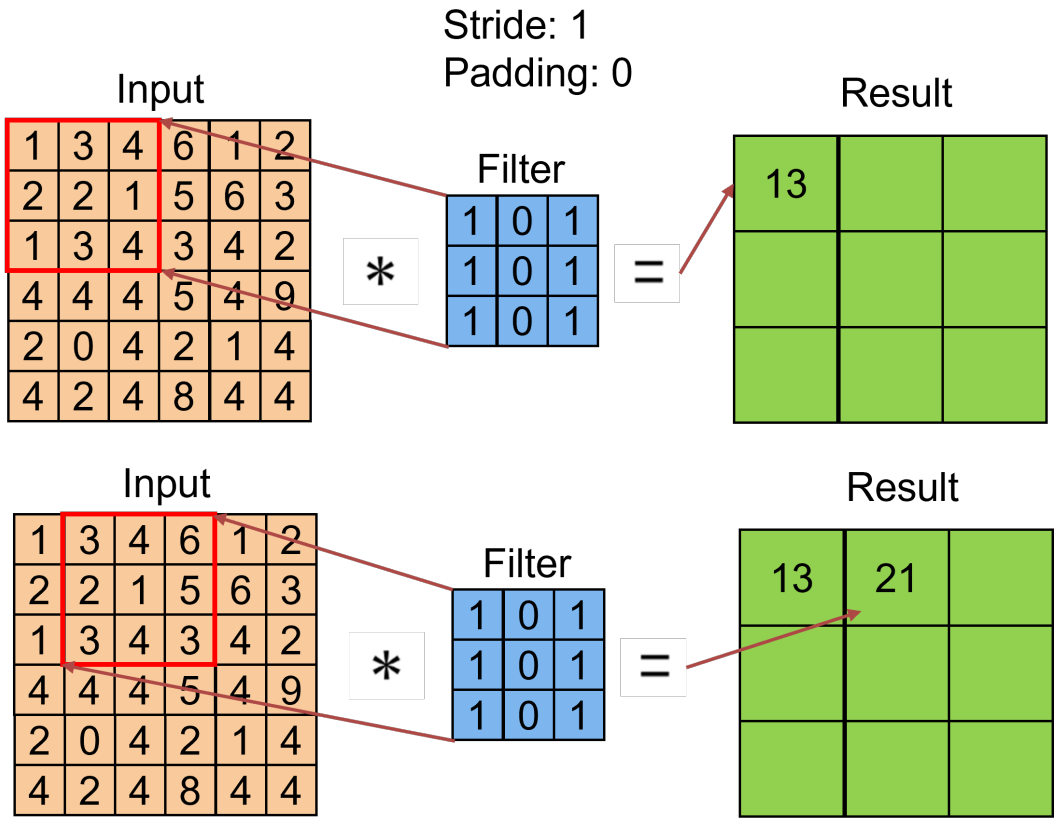


Figure 2.1: Convolution operation. Convolutional kernel performs the operation by sliding from the input. The padding and size difference between input and output is shown at the top.

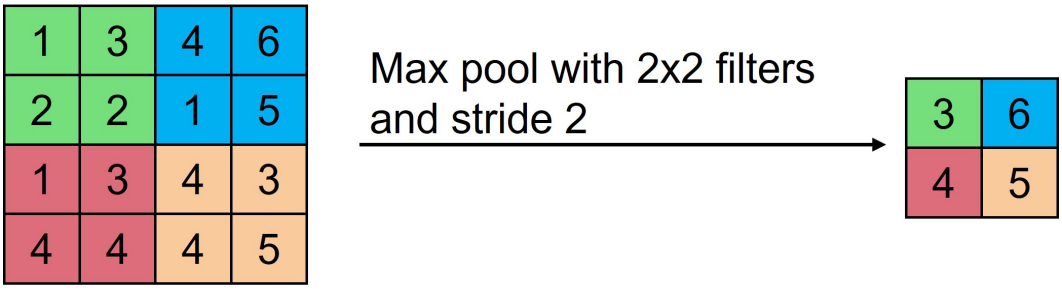


Figure 2.2: Pooling operation. After performing the convolution, the pooling operation is applied to downsample the size of feature map while keeping the significant activation values.

of the object we want to classify may contain many other objects. For example, a cat that appears somewhere in the car image may mislead the classifier. Pooling helps to alleviate this phenomenon and promotes CNN.

- Normalization Layer

One of the main challenges of CNN is the vanishing gradient, which makes training unstable. Ioie and Szegedy [32] found that this is mainly due to changes in internal covariates, which are caused by changes in the data distribution through the information forwarding in CNN. They proposed the technology of batch normalization to standardize each batch of images to get zero mean and unit variance. There are several types of normalization, batch normalization (BN), layer normalization (LN), group normalization (GN), etc. They have their own scenario that can improve CNN the best. For example, if the batch size is set to a small value, normalization layer could not extract reliable mean and standard deviation to represent the whole dataset, leading to useless normalization. Given the dataset with samples x_i, the mean value and variance are represented by μ_β and σ_β, the batch normalization performs the following equations.

$$\mu_\beta = 1/m \sum_{i=1}^{m} x_i \tag{2.1}$$

$$\sigma_\beta^2 = 1/m \sum_{i=1}^{m} (x_i - \mu_\beta)^2 \tag{2.2}$$

$$\hat{x} = x_i - \mu_\beta / \sqrt{\sigma_\beta^2 + \epsilon} \tag{2.3}$$

$$y_i = \gamma \hat{x_i} + \beta = BN_{\gamma,\beta}(x_i) \tag{2.4}$$

, where the first two steps are used to calculate the mean and standard deviation values from each batch size. Then we will normalize the inputs x_i using the calculated mean and standard deviation, followed by scaling and translating operations from the third equation.

- Dropout Layer

Overfitting is a phenomenon in which the network performs well on the training set but not on the test set. This is usually due to excessive reliance on specific features that appear in the training set. Dropouts are a technique to suppress overfitting. It can randomly set some activation values to 0 to avoid overfitting. The network has to explore more paths to classify images instead of relying too much on certain features. Dropouts are one of the key elements in AlexNet.

2.1.2 Activation Function

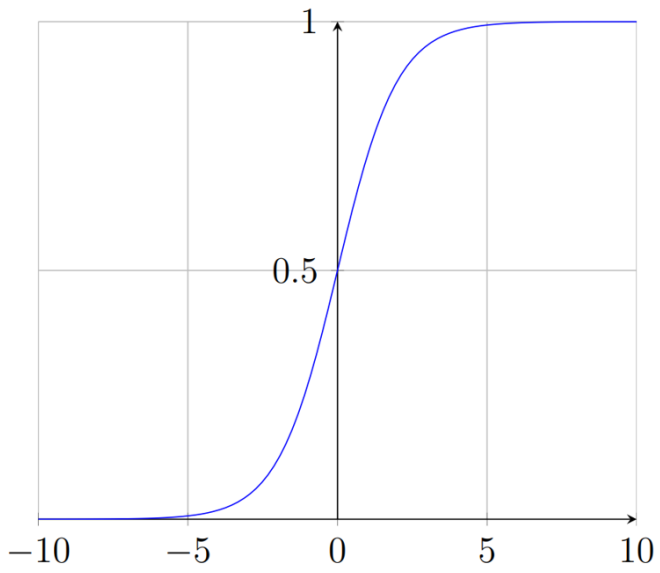


Figure 2.3: Sigmoid activation function reprtinted from [62]

The activation function is applied to the neurons of the artificial neural network. It conducts the nonlinear mapping between the input and the output of the neuron. It calculates the inner product of the input vector and the weight to produce a real number. The real number will be added as an offset term, and it is derivable so that it will be used in the following backpropagation optimization process. The purpose of the activation function is to map the linear transformation into a high dimensional nonlinear space [29, 83]. Typical activation functions are Sigmoid, Rectified Linear Activation Unit (ReLU), Leaky ReLU, etc.

The sigmoid function is monotonic, shown in Figure. 2.3 It is constrained by a pair of horizontal asymptotes ∞ so that it has an "S"-shaped curve. The physical meaning of a sigmoid's output is close to the behavior of biological neurons, that is, the output range $(0, 1)$ can be expressed as a probability. Also, the output can be easily applied to the cross-entropy loss function without any other modifications. Also, the sigmoid is a soft saturation activation function. Specifically, the derivative of infinity input will be close to 0, leading to gradient vanishing problem. It is a major obstacle to achieve a stable training process.

To solve the gradient vanishing problem, the ReLU is proposed and widely used in the deep learning community. As shown in the Figure. 2.4, the ReLU avoids the gradient vanishing problem by squashing the input value between 0. When the input value smaller than 0, the output will be saturate, and it has identical continous derivate when the input value bigger than 0. The appearance of ReLU is the key to gradient vanishing problem and boost the performance of the very deep neural network.

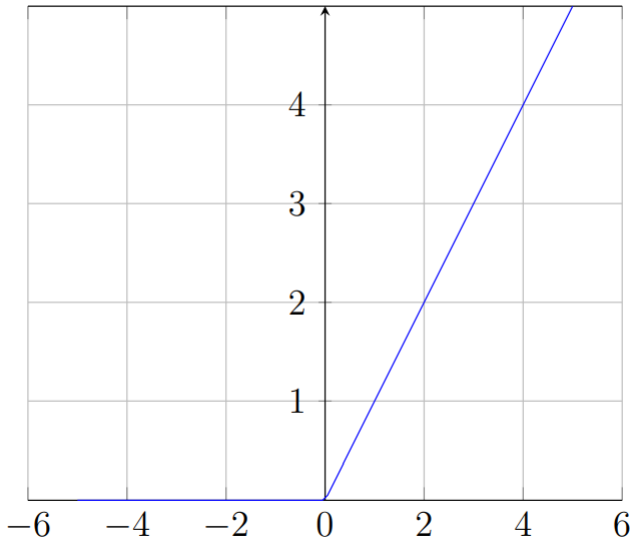


Figure 2.4: ReLU activation function reprinted from [62]

2.1.3 Loss Function

The loss function sets up an objective to optimize the machine learning method to converge into the global/local optimal point. Specifically, it evaluates how well the method performs

by calculating the error between the prediction and the ground truth. The smaller the loss is, the more accurate prediction of the model made. Furthermore, it provides the direction of optimization for the back propagation process. The gradients will be generated by the loss function and the weights of model will be updated. Generally, the category of loss function can be divided into two folds, regression loss and classification loss.

Regression loss. In the field of regression task, the L1 loss and L2 loss are commonly used. As shown in 2.5, L1 measures the average sum of the absolute differences while the L2 measures the average squared differences between predictions and ground truth labels. In this work, we formalize the anticipation task as a real time remaining time regression task. Therefore, the L1 is mainly used in this work.

$$L1 = |f(x) - y| \tag{2.5}$$

$$L2 = |f(x) - y|^2 \tag{2.6}$$

, where the $f()$ denotes the function modeled by the neural network, x is the input of the network and y is the ground truth.

Classification loss. For the classification losses, the cross-entropy loss is the most common setting for classification problems. As shown in Eq. 2.7, it measures the performance of a classification model whose output is a probability value between 0 and 1. As the predicted probability distribution is closer to the real probability distribution, the loss value gets lower. This indicates that the model is performing better in terms of classification. In this work, we applied an auxiliary classification task as the regularization term and cross-entropy is utilized.

$$CrossEntropy = -1/N \sum_{i} \sum_{c=1}^{M} y_{ic} log_{p_{ic}} \tag{2.7}$$

, where the N is the number of the training samples, M is the number of the categories, the p_{ic} is the prediction from the neural network and $y_(ic)$ is the ground truth.

2.1.4 Other Features

In addition to the RGB feature from the raw image, there exists many other features we can use to improve the performance of machine learning task. The source of these features can be from pre-trained neural network, conventional computer vision approaches and so on. For example, the SFTGAN [79] utilizes the semantic feature map from the segmentation map to modulate the model's parameter by recognizing the regions with respect to different semantics. Therefore, this feature explicitly tells the model which region belongs to which class. In the other works [67, 69], the optical flow is widely used to provide sequential movement information. The previous works show that the optical flow contains more temporal information compared to the RGB feature. It is extracted by calculating the movement of the pixels from adjacent RGB frames and it is extremely useful for the video-based analysis. It calculates the spatial movement of each pixel in the image, offering the clue to object's transformation. Furthermore, Junwei et al. [47] exploit the use of detected bounding boxes to infer the interaction relation between different objects. It considers the euclidean distance along the spatial coordination, and also be aware of the size difference among objects. This prior alleviates the intermediate bounding box detection and allows the network to understand the interaction between the subject and the objects and predict the trajectory. A more complex example is the generative models, such as VAE [41]. VAE is responsible for generating a feature vector distribution and sample from that. The sampled vector will be used as the input of next decoder to generate the final output. In this thesis, we incorporate the segmentation map, detection boxes and binary presence signal to boost the model's performance.

2.2 Interventional Health Care Applications

Endoscopic surgery is a minor surgery which allows the surgeon to observe the inside of body and perform the medical procedure without major intrusion. Surgeons utilize the scope which is a flexible tube with a camera and light at the tip to inspect the anatomical structure and utilize the the instruments to perform surgical actions, e.g., dissection, clipping, cutting. This surgery provides an easier recovery, less pain and discomfort for the patient.

As shown in the Figure. 2.5, endoscopic surgery, termed as surgical workflow, has a relatively

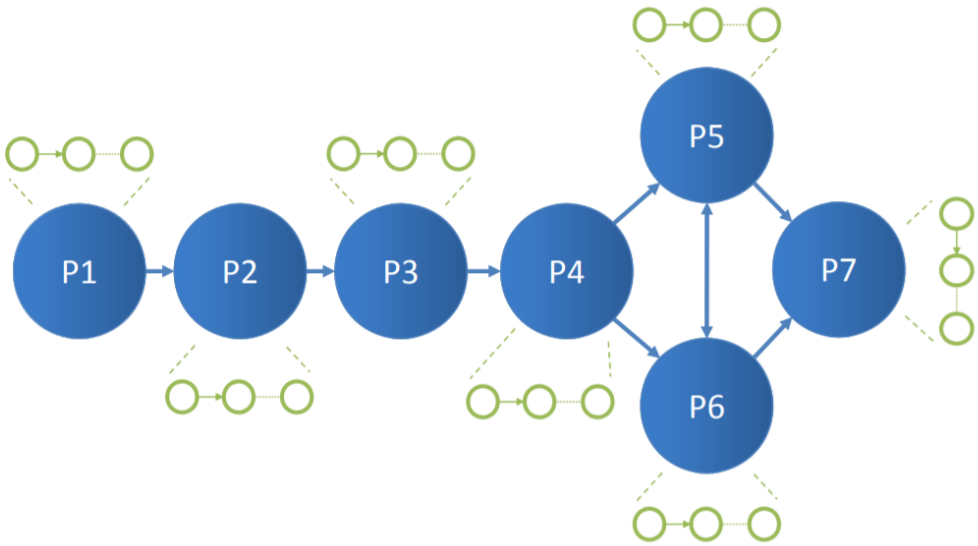


Figure 2.5: Surgical phase transition reprinted from [74]. The surgery can be partitioned into multiple phases and each phase is responsible for one clinical operation.

fixed order which the surgeon can follow consecutively. Therefore, a prominent application called phase recognition is proposed and explored as the main research direction. Also, skill assessment that assesses how well the surgeon handles the surgery is appealing since it provides useful information for future decision-making, such as readmission preparation. Unlike the recognition focusing on what is going on, the anticipation of workflow aims to predict what will happen for the future instrument preparation and many other applications. In the following sections, we will discuss these three applications and their state-of-the-art deep learning methods.

2.2.1 Surgical Phases

Firstly, we will talk about details in gallbladder retraction surgery by defining multiple phases. As shown in the Figure. 2.5, the surgical workflow can be classified into different phases and here are the definitions of them.

Phase 1. Phase 1 is officially termed as the Preparation phase, where the surgeon will estimate the status of the patient by placing the grasper into the human body. This phase is usually shorter than the other phases and can only happen at the start of the surgery. As shown in the Figure. 2.6, when the grasper holds the gallbladder for a certain time, the surgeon will place

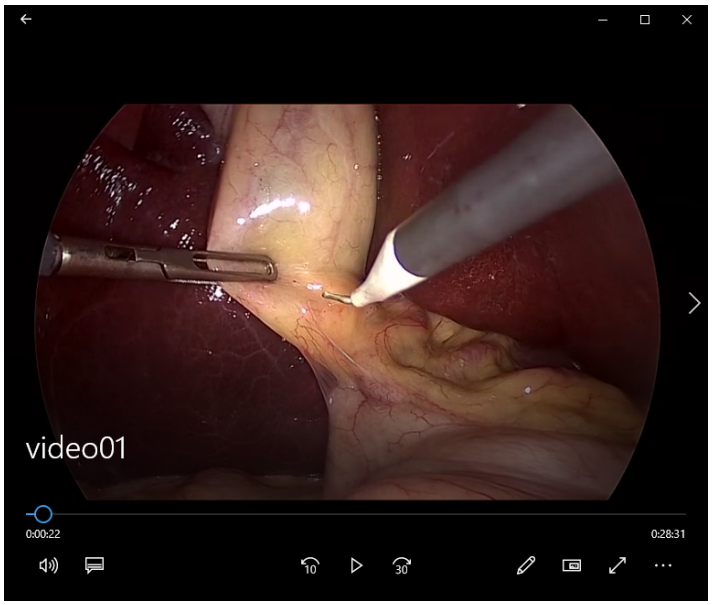


Figure 2.6: Selected figure of Phase 1 from Cholec 80 dataset.

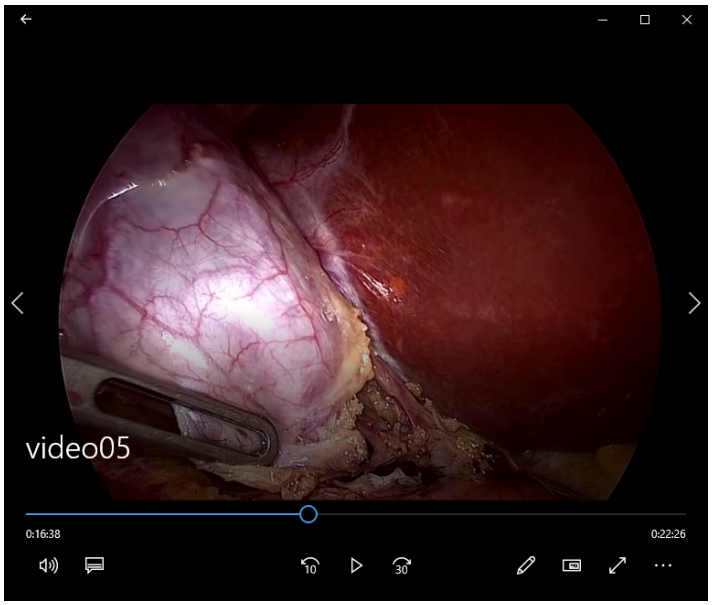


Figure 2.7: Selected figure of Phase 2 from Cholec 80 dataset.

the hook inside, indicating the end of phase 1.

Phase 2. Phase 2 is called the CalotTriangleDissection phase, aiming to dissect the gallbladder from the liver using the hook and grasper. It completely dissects the Calot's triangle area, which must not contain more than one biliary tract element. During the dissection, it should always keep in contact with the gallbladder by using the grasper. When the triangle area is

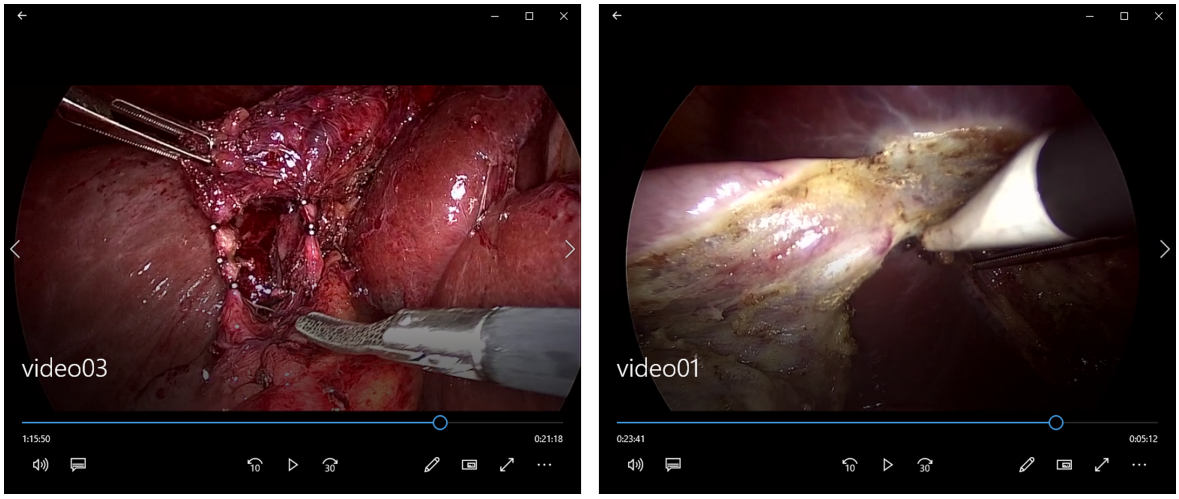


Figure 2.8: Selected figures of Phase 3 and phase 4 from Cholec 80 dataset.

clear, phase 2 is approaching the end, and phase 3 is going to happen, as shown in the Figure. 2.7.

Phase 3. Phase 3 is called ClippingCutting, which is to clip the cystic artery and duct by the clipper and cut them using the scissor. Normally, the cystic artery and duct are attached tightly together, and a single clipping and cutting sequence is required. As shown in the Figure. 2.8 on the left, when finishing the cutting, the surgery will go into the next phase.

Phase 4. After cutting the cystic artery and duct, the gallbladder should be easily removed. Thus, the hook will be introduced again to dissect the connected areas between the gallbladder and liver. This phase is called GallbladderDissection. This phase will last until the gallbladder is fully detached from the liver, as shown in the Figure. 2.8 on the right. When the last tissue connecting the gallbladder and the liver is severed, this phase ends, and the next phase starts.

Phase 5. GallbladderPackaging, as the phase 5, aims at packaging the removed gallbladder into the specimen bag. During this phase, the specimen bag will be introduced firstly, and multiple graspers will be utilized to operate the specimen bag bag and move the gallbladder, as shown in the Figure. 2.9

Phase 6. Phase CleaningCoagulation is an important phase because this is highly correlated to the patient readmission rate [75] and can handle many complications in this phase. Before this phase, the surgeon observes the inner status and decides if this is an emergency situation

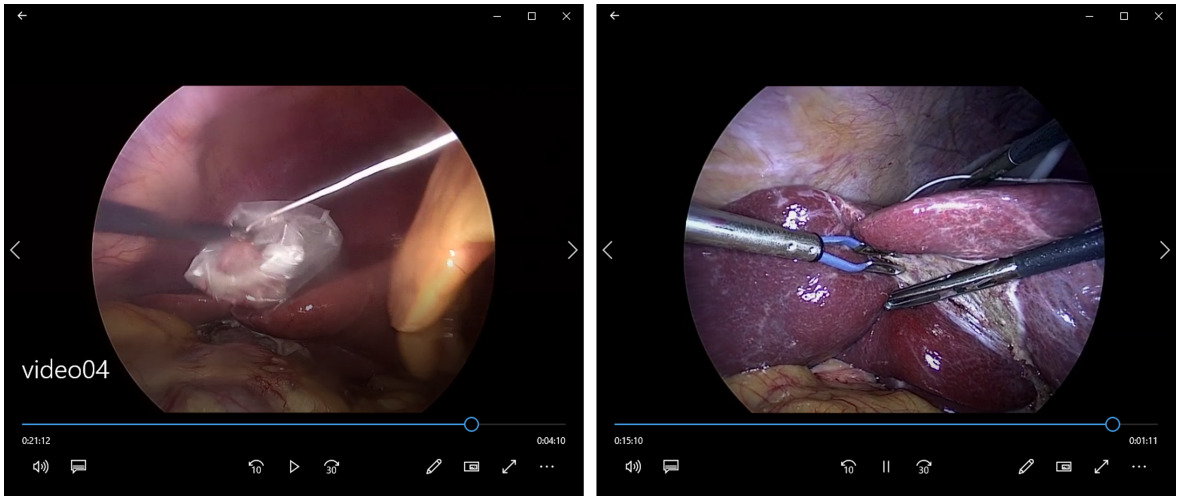


Figure 2.9: Selected figures of Phase 5 and phase 6 from Cholec 80 dataset.

that needs the bipolar to be placed. If so, the surgeon will introduce the bipolar and stop the bleeding, as shown in the Figure. 2.9.

Phase 7. This is usually the last phase of the whole surgery and surgeon will completely remove the gallbladder from the body.

2.2.2 Phase Recognition

In the community of computer-assisted interventions (CAI), recognition of the surgical workflow's phases is an important topic because it offers solutions to numerous demands of the modern operating room (OR) [8, 74]. Phase recognition is essential for context-aware systems design and benefits many decision-makings, such as OR optimization, surgical process monitoring, and automated assistance. Specifically, phase recognition needs to classify each frame of surgical video into a category. The category is defined as surgical phases in Figure. 2.5 or surgical actions in the work of [55].

The recognition methods can be divided into offline [74] and online [34] manner. The former one determines the current frame's category conditioned on the past and the future, while the latter one only allows the model to observe the past frames before recognition. To apply the model in the online clinical scenario is challenging. Firstly, it needs the model to be efficient enough that can process each frame within one second, because we resample the video into 1 frame per second as discussed in 4.1. This is tough when using hand-crafted features, such

as intensity and gradient as visual feature extraction backbone [34]. Thanks to the automatic feature extraction in deep learning, this has been greatly solved but the training cost is another challenge to us. Novel works utilizes ResNet [27] and LSTM [30] jointly trained. However, this training strategy will need more computational cost, leading to a small receptive field and thus degrading performance. In the work of [10], it opts for the MS-TCN [14] with a dilation factor to solve the training problem and achieves a state of the art performance in the online recognition.

Also, diverse attempts have been made for the phase recognition by various types of features. In [6,17,38,57], binary tool presence signals are used and form a multi-task learning strategy. The Yueming et al. [34] utilizes the RFID tags acquired from sensors to help the surgical recognition.

2.2.3 Skill Assessment

Safe surgical care is not always accessible for five billion people in the world [3]. According to the World Health Organization, up to 25% of patients undergo the operations that require a stay in the hospital and suffer complications [33, 82]. Since the surgeon skill varies dramatically, poor performance of surgery will make patients suffer from the complications and harm. Therefore, an adequate training and automated coaching by surgical skill assessment [24, 28, 37] becomes the next topic of the surgical workflow analysis.

Yixin et al. [22] firstly release the the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) for this field. The whole dataset contains three surgical tasks, Suturing (SU), Knot-Tying (KT) and Needle-Passing (NP) in the da Vinci surgical system. Each task is represented by a short video where the surgeon performs the surgical procedure by system. The da Vinci surgical system gives the surgeon an advanced set of instruments to use in performing robotic-assisted minimally invasive surgery. The JIGSAWS also includes manual annotation for surgical technical skill. An experienced surgeon will watch the video and assign a global rating score (GRS) using a modified objective structured assessments of technical skills (OSATS) approach [53]. Inspired by this dataset, statistical analysis is explored to describe the skill levels. For instance, [36, 48, 72] compute movement time

as feature to discriminate the skill level. Other novel measures of motion, such as energy expenditure [59], semantic labels [13], tool orientation [68], and force [72], can also provide discriminative information in measuring skills. As the deep learning shows its feasiblity in automated feature extraction, [80, 84] starts to utilize deep learning method to evaluate the objective skill of surgeon. This has achieved a significant improvement compared to the previous hand-crafted feature based methods, and a lot research effort has been put in this direction.

2.2.4 Instrument and Phase Anticipation

Different from the surgical workflow recognition which focuses on the post-hoc analysis, the anticipation task focuses on predicting the occurrence of instrument or phase ahead. This provides a scope of the future and improves intra-operative patient care. The anticipation can be classified into instrument anticipation and surgical phase anticipation, are rarely explored by current works. Though, there are few attempts that handle the anticipation as a specific predictive task [63, 75]. There are also other works [21, 39] adopting the segmentation prediction idea from the conventional vision field to handle the anticipation. It can predict roll-outs of the surgical process in the form of discrete label sequences for operative phases over time. However, their methods can only anticipate 15 seconds of future which limits its application in real world. Also, thanks to the generative adversarial networks [25], Yutong et al. [5] uses an encoder-decoder structure based to predict multiple faithful future possibilities. It handles the uncertainty by sampling multiple times from the output of the encoder.

2.2.5 Evaluation Metrics

Recognition. As mentioned above, three main tasks of the surgical workflow have their corresponding strategy when applying deep learning methods. The phase recognition is usually transformed into a temporal segmentation task that performs frame-wise classification. Therefore, the classification-based metrics, i.e., accuracy, recall, precision, will be the best candidate to evaluate the model's performance. Specifically, we can also calculate class-wise metrics to evaluate how well the model performs on each class. This provides a detailed view to find where the problem hides in the model and fix it.

Skill Assessment. Similar to the phase recognition task, the skill assessment task is mostly evaluated by the classification-based metrics. The difference is that the skill assessment task is a video-wise classification problem that assigns each video a score indicating how well the surgeon performs during the surgery. However, due to the unbalanced distribution in the surgical dataset, a metrics like Average Precision (AP) is favored for small dataset. AP measures the performance of the learned model in each category, and Mean Average Precision (mAP) evaluates the performance in all categories by averaging all APs.

Anticipation. Compared to the previous tasks, the instrument and phase anticipation is more complicated and various. It can be regression-based metrics or classification-based metrics depending on the specific task we designed. Here we transform the anticipation task into the remaining time estimation problem and investigate its suitable evaluation metrics. The most straightforward metric is mean average error (MAE), which calculates the frame-wise euclidean distance between two time series. This generally indicates the overall difference and provides a coarse idea about how close two sequences are. Furthermore, as instruments are not always predictable, making precision metrics popular, researchers propose the precision average mean error (pMAE) for evaluating the anticipation task [21, 21, 78].

2.3 Spatial Feature Extraction

Since AlexNet [43] made a blockbuster in the ImageNet competition in 2012, deep learning has entered a stage of vigorous development. Convolutional neural networks adaptively extract a feature vector for the image, leading to an unprecedented power in the image field, such as image classification, target detection, positioning, and semantic segmentation. Since the development of convolutional neural networks, there have been many variants, each with its own characteristics. Based on the most basic classification network, this section summarizes various classic deep neural networks.

2.3.1 LeNet5

In 1998, LeNet [46] is firstly proposed to handle the handwritten number recognition and it proposes to use three architectural ideas to ensure a certain degree of invariance to translation,

scaling, and distortion of the image by designing the network with Local receptive fields, weight sharing and downsampling operations.

LeNet5 uses a convolution-downsampling-nonlinear activation sequence (downsampling uses average pooling), and this network feature continues to this day. In addition, fully connected layer and feature maps are also the basic components of for the convolutional network construction. It can be seen that LeNet5 has made much contribution to the convolutional network. However, due to the limitations of data and hardware conditions, deep learning remained silent until 2012.

2.3.2 AlexNet

In 2012, AlexNet [43] was born, and it greatly surpassed the second place player in ILSVRC-2012. Since then, deep learning has entered a stage of rapid development. AlexNet and LeNet are similar in overall architecture, but AlexNet is deeper and uses more techniques. Firstly, it utilizes the ReLU as the nonlinear activation function. It also emphasize the effect of the fully connected layer for image classification. This work innovatively proposes the dropout operation to reduce the non-trivial overfitting problem during the training. The data augmentation is applied as well. The network is trained on two GPUs, leading the development to the future grouped convolution.

2.3.3 VGG

VGG [70] is one of the most commonly used basic networks for the current research. The reason is that the network structure is very simple and unified. It is a stack of convolution layers and pooling layers. However, one of the biggest drawbacks is that the amount of parameters is relatively large. An improvement of VGG compared to AlexNet is to use consecutive $3x3$ convolution kernels to replace the larger convolution kernels in AlexNet, i.e., $11 \times 11, 7 \times 7, 5 \times 5$. For a given receptive field, using a stacked small convolution kernel is better than using a large convolution kernel, because multiple non-linear layers can increase the depth of the network to ensure more complex learning, and the computational cost is relatively small.

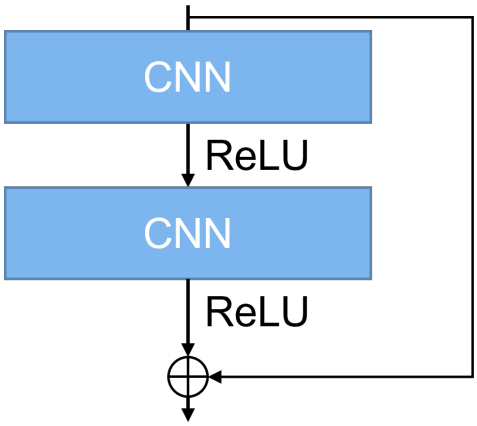


Figure 2.10: Basic block of ResNet. The ResNet is stacked with multiple basic blocks, each of them is composed by two convolutional layers and skip connection. Figure reprinted from [27].

2.3.4 ResNet

In the development of deep learning, from LeNet to AlexNet, to VGG, the depth of the network is constantly deepening. Experience has shown that the depth of the network has a crucial influence. A network with deep layers can extract the low-level, middle-level and high-level features of the image. But when the network is deep enough, just stacking more layers at the back will cause many problems: the first problem is vanishing / exploding gradients, back propagation cannot effectively update the gradient to the previous network layer, making the previous layer parameters to fail to update. The second problem is the degradation problem, that is, when the number of network layers is stacked, it will cause optimization difficulties, and the training error and prediction error will be greater. Note that the greater error here is not caused by overfitting.

ResNet [27] aims to solve the phenomenon that the difficulty of training increases after the network is deepened. It proposes the residual module, which contains two 33 convolutions and a shortcut connection, as shown in Figure. 2.10. Shortcut connection can effectively alleviate the disappearance of gradients caused by excessive depth during backpropagation so that the performance of the network will not deteriorate after the network is deepened. In addition to computer vision, short-circuit connection is also used in the fields of machine translation and speech recognition/synthesis. In addition, ResNet with short-circuit connections can be

seen as an integration of many networks with different depths and sharing parameters, and the number of networks increases exponentially with the number of layers.

2.4 Anticipation Methods

In the surgical workflow field, the anticipation problem is not fully investigated and receives less attention. Therefore, we will introduce some works for anticipation tasks from two perspectives, the future prediction and action anticipation in conventional computer vision.

Future prediction aims at predicting diverse representations, tools presence [63], short-term pixel-level [50] and long-term high level [77]. In [77], they propose to learn a high-level pose representation and predict the low-level pixel representation based on observing the learned high-level structure, which avoids the error propagation happening in recurrent networks facing the low-level prediction. The predictions generated by previous works could be further fed into a classifier for action/phase recognition, which is the goal of the anticipation. In [19], they explore an adaptive state-transition model by considering the history of the current work step to predict the next phase in surgery. However, it is a non-differentiable method and not applicable to frame-wise prediction for surgery video. Besides, most works [65] are focusing on the short time videos (10 seconds), which is not robust given the long-term laparoscopic video (20 to 60 minutes). And they are not able to both predict the phases and their start time, end time, duration at once [2].

2.5 Temporal Modeling Methods

2.5.1 Markov Chain

The surgical video is a sequential data and it can be represented by a set of hidden states. In detail, the hidden states are the surgical phases defined by the experts and there exists a transition matrix among the phases. Therefore, the most straightforward method to model this scenario is the markov chain. The markov chain is a sequential model which strictly assumes that the future states are not dependent upon the previous steps that led up to the present state. Different from the general stochastic process, the markov is memory less. Its property points out that the probability of future states prediction can only be influenced by

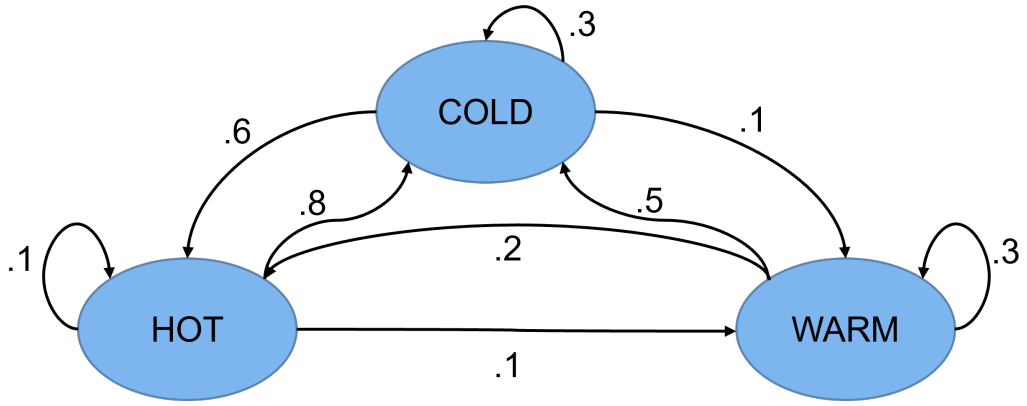


Figure 2.11: Hidden markov model. Each state is influenced by its previous state [60]. In the surgical scenario, each state could be represented by a surgical phase.

the current states. This is a general property that many applications are following in the real life.

As shown in the Figure. 2.11, it's as if to predict tomorrow's weather, you could examine today's weather, but you weren't allowed to look at yesterday's weather. The probability distribution of state transitions is typically represented as the Markov chain's transition matrix. If the Markov chain has N possible states, the matrix will be an $N \times N$ matrix, such that entry (I, J) is the probability of transitioning from the state I to state J. Additionally, the transition matrix must be a stochastic matrix, a matrix whose entries in each row must add up to exactly 1. This makes complete sense since each row represents its own probability distribution.

2.5.2 Recurrent Neural Network (RNN)

Recurrent neural network (RNN) is a type of artificial neural network that can create connections between nodes in the network graph by introducing weights for each node and maintain an internal state. The benefit of adding states to neural networks is that they can explicitly learn and use context in sequence prediction problems. Different from the feed-forward neural networks, RNNs can use their internal state to process variable length sequence of inputs [1, 12]. This allows it to exhibit temporal dynamic modeling and applicable to tasks such as speech recognition [66], sequence prediction.

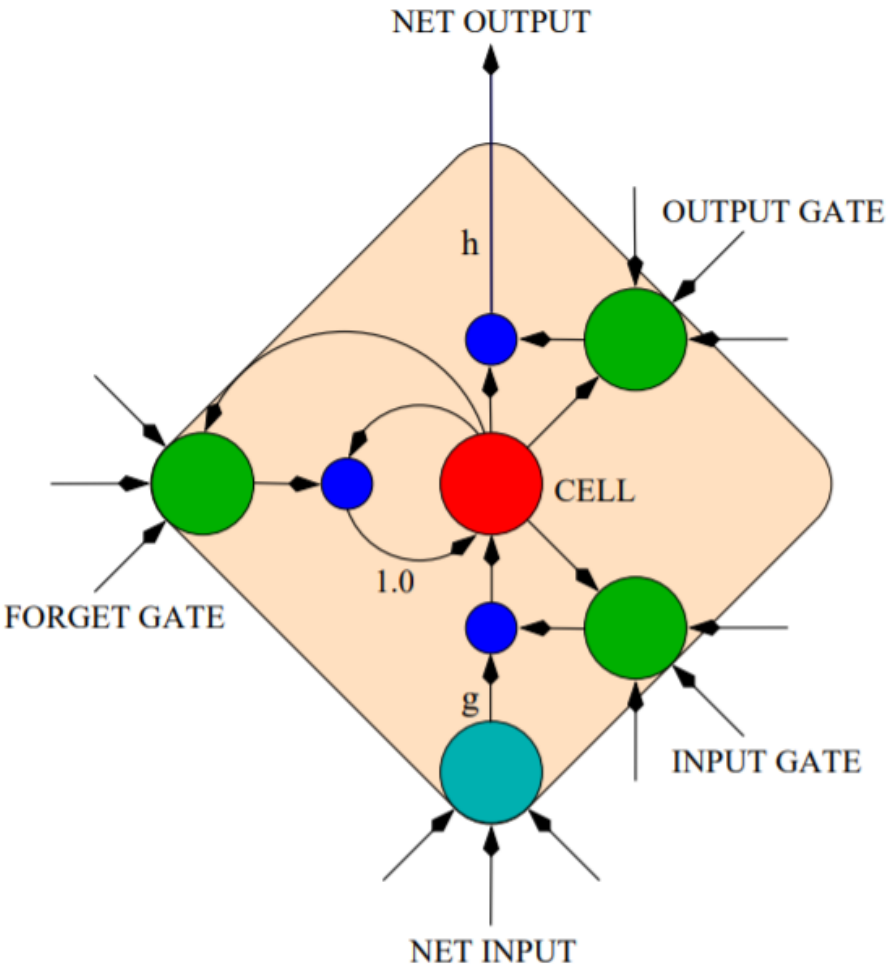


Figure 2.12: LSTM cells. LSTM is built by multiple cells, which has forget gate to discard the information, the input gate is to introduce the new input and the output gate to propagate the information. Figure reprinted from [26]

RNNs come in many variants among which the most popular and widely used is Long short-term memory (LSTM). The development of LSTM successfully resolves the gradient vanishing problem when modeling long-range time series. Also, its gate design helps to keep the long-term dependency into consideration for the time series prediction. LSTM as an artificial RNN architecture is widely used in the field of deep learning, especially in NLP [16], video understanding, and anomaly detection in spatial-temporal network traffic data.

As shown in Figure. 2.12, the basic unit of LSTM is a cell, which contains input gate, output gate and forget gate. These three gates control the information flow into and out of the cell to decide whether to remember the information from the recent/distant past. Also, this

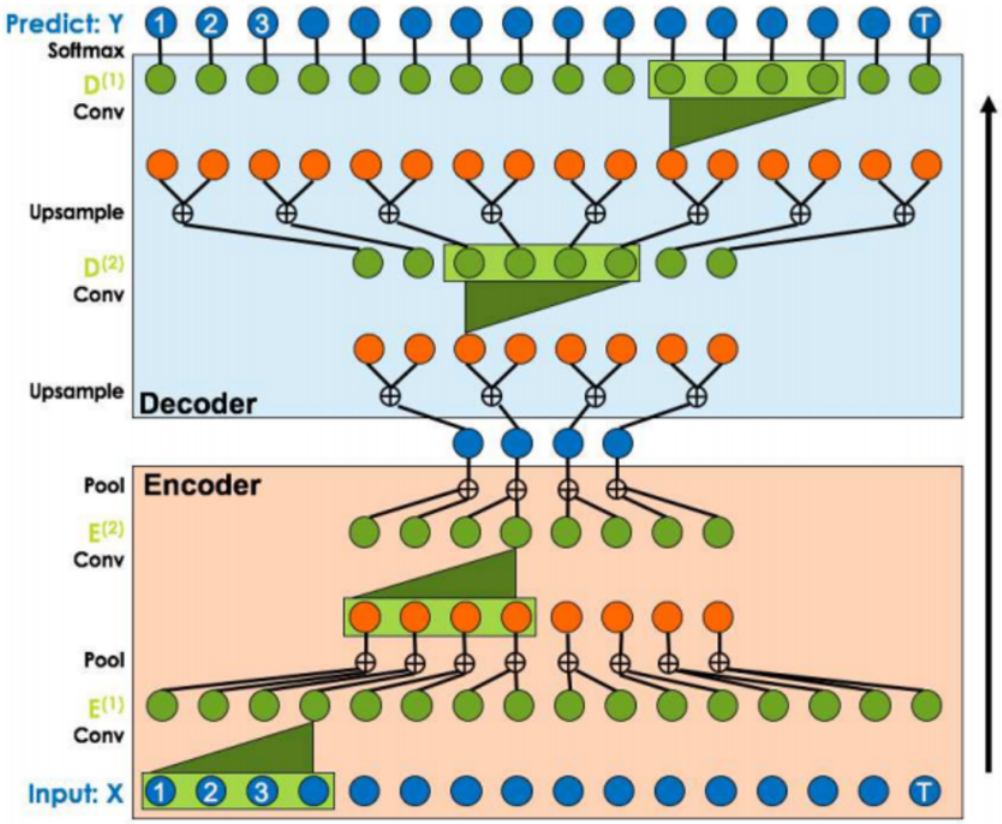


Figure 2.13: Structure of multi-stage convolutional network. The network consists of the encoder for the representation learning and the decoder for the sequence prediction. Figure reprinted from [44].

characteristic allows LSTM to outperform the traditional RNNs with an obvious gap. Also, the LSTM can handle variable length of time series with a moderate computational cost.

2.5.3 Temporal Convolutional Neural Network (TCN)

The success of convolutional neural network (CNN) in image classification tasks encourages researchers to explore CNN's modification in sequential tasks. In 2016, Lea et al [44] firstly proposed and prove that CNN can be a valuable tool for sequence modeling and forecasting, give the right modification. Therefore, Temporal Convolutional Networks (TCNs) are applied and achieve an exciting result for video-based action segmentation. Unlike the RNNs modeling the sequence iteratively, the TCN takes the whole sequence as input at once. This makes TCN more efficient than the RNNs and is more feasible to real-world applications. Also, the TCN is fully implemented by convolution layers which is consistent to the CNN,

leading to a unified network structure when handling the video classification task. The component of TCN can be classified into causal convolution and non-causal convolution. Causal convolution means the current prediction only depends on the elements that occur before and vice versa for non-causal convolution. The typical TCN encoder-decoder structure is shown in Figure. 2.13, it takes a series of any length and outputs an output with the same length.

2.5.4 Dynamic Time Warping

Dynamic time warping (DTW) is one of the popular algorithms in time series analysis. It is a template-based algorithms, which needs the reference sequence and the source sequence. The DTW has been widely applied in the speech recognition [35], online signature recognition and partial shape matching. It helps to resolve the speeding rate difference problem in the video, audio and graphics data. Given two temporal sequences vary in speed, the DTW can warp the source sequence to the reference in the time dimension by producing a discrete matching between existing elements of one series to another. Specifically, the DTW calculates the optimal match between two sequences. The optimal match is the match that can warp the source into the reference with the minimal cost. The cost can be defined as the sum of distances, e.g. euclidean, vector distance. By using the dynamic programming algorithm, we can find the optimal match for the source-reference pair. The non-linearly warped sequence possesses the similar pattern as the reference sequence, leading to a heuristic temporal modeling method. Also, the cost of that optimal match provides the distance-like quantity between given sequences and is useful for the sequence classification tasks. Though the DTW is effective and efficient to compute, it is not differentiable and can not be applied to the deep learning framework due to its template-based characteristic. Recently, [49] proposes to integrate warping mechanism to shrink the intra-classes difference and enlarge the inter-classes difference. More variants need to be explored in the future.

2.6 Multi-Task Learning

Instead of learning a single network for each task, we can design a multi-task learning schema with one shared backbone and multiple prediction heads to handle different tasks. This is especially helpful when the tasks are highly correlated. By performing the multi-tasking,

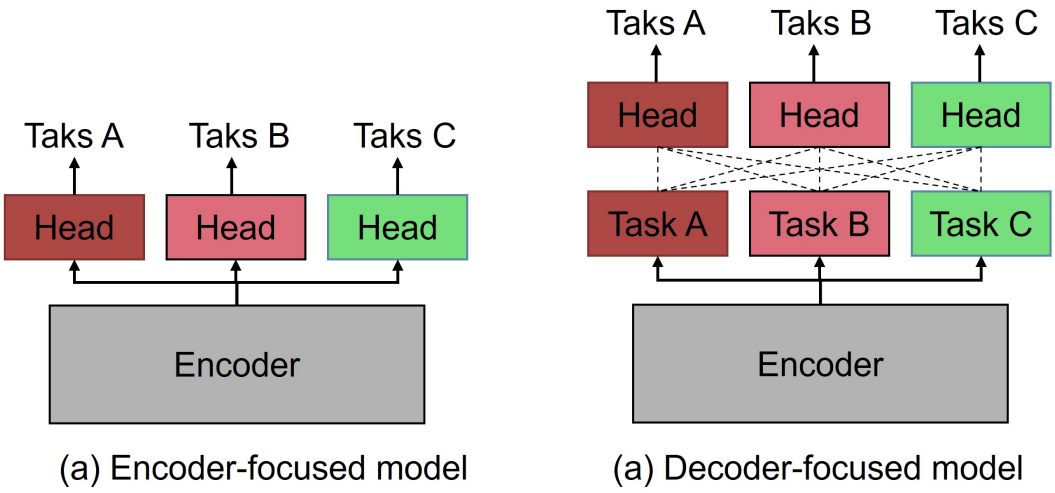


Figure 2.14: Multi-task learning strategy. Encoder-focused strategy aims to fuse the information in the early stage and the decoder-focused model performs the late-fusion. Figure reprinted from [76]

there is no need to train separate models for different tasks, which could largely save the computational cost. According to the [76], the approaches of deep learning based multi-tasking methods can be divided into encoder-focused and decoder-focused architectures.

2.6.1 Encoder-focused model

As shown in the Figure. 2.14, the encoder-focused architectures handle multiple sub-tasks using the same encoding stage. The encoder is also called backbone network. Specifically, previous works [54, 71] share the same backbone network that extracts the shared feature from the input. Then multiple independent task-specific prediction heads will be added to get task-specific predictions. Instead of learning a representation for each task, it outputs the general representation optimized by multiple tasks. This is called hoc sharing strategy to combine different tasks. Furthermore, researchers have focused on where and how the feature sharing should happen in the encoder. Different sharing strategies can be found in [76].

2.6.2 Decoder-focused model

Different from the encoder-focused architectures, decoder-focused model makes the initial task predictions to generate task-specific features. Then features from the initial prediction

will be combined in an on-off or recursive manner, which improve the final performance of each task. As shown in Figure. 2.14, the middle layer of prediction acts as an weak supervision which regularizes the task-specific features to be more informative and provides information exchange strategy for the final prediction. Encoder-focused architectures follow a pattern that directly predict all task outputs from the same shared feature. The predictions are generated in parallel and can not be improved furthermore by the other task outputs. This makes it unable to capture the commonalities and differences among tasks. Thanks to the decoder-focused approaches that connect different tasks together, multi-task learning exchange the information during the decoder stage. This helps the model alleviate the issues mentioned above and the performance gets boosted.

Chapter 3

METHODOLOGY

Our work is composed of two parts, a spatial feature extractor to extract a single feature vector for each frame and a temporal model using causal MS-TCN. During training, these two components are trained in a two-step training manner. For an online video, each frame will be sequentially fed into the feature extractor and then appended with the previous feature vectors to form a sequence. Then, our causal MS-TCN will take the sequence as the input and output a result sequence with same length as the input. Therefore, the last element from the result sequence is the prediction for the current timestamp, and the previous predictions can be stored offline or removed.

In this work, we follow two steps to handle this task:

- Spatially, the feature extractor models the surgeon's intention through extracting rich geometric interactions of the instrument-instrument and semantic interactions of instrument-surrounding. This is implemented by our proposed instrument-instrument module (IIM). Also, motivated by the recognition methods [33, 34, 56, 74], we introduce tool and phase presence signal to indicate the current tool/phase's occurrence.
- Temporally, we utilize causal dilated MS-TCN [14] to capture long-term dependencies with a large receptive field. For the current feature vector, our causal MS-TCN predicts its outputs by observing only the previous frame's feature vectors, not the following ones. This restriction makes our system feasible for the online surgery scenario.

To effectively capture the instrument-surrounding and instrument-instrument relations mentioned above, we explicitly design a instrument interaction module (IIM) with two branches:

- Instrument-Surrounding is used for capturing the relation between surgical instrument and its surrounding anatomical structures. It utilizes the pre-extracted segmentation map to indicate the region of interest and tells the location of the instrument.

- Instrument-Instrument is used for capturing the instrument action that surgeon makes. For each frame, we utilize the pre-trained tool detection model to extract tool bounding boxes' coordinates. Then, we follow [47] to build a feature vector for each tool combination. The vector that consists of the translation difference and the size relation is the powerful prior for estimating the surgeon's intention.

3.1 Task Formulation

There are two appoarches to handle the anticipation task, dense segmentation prediction [2] and remaining time [63] regression. Dense segmentation prediction method outputs a 0/1 sequence, indicating the non-presesnce/presence of the specific surgical instrument/phase. The remaining time prediction method outputs the remaining time until the next occurrence of the specific instrument/phase. Here we opt for the latter one because the output from dense segmentation prediction contains many short segmentation, shown in Figure. 1.1. It is ambiguous and ineffective for the downstream applications, such as the surgeon's decision making. Our IIA-Net handles anticipation as a real-time remaining time regression problem without any latency or pre-loading process. Also, we add a classification regularization term to improve the model's performance.

3.1.1 Regression

We process the anticipation task as a regression problem both for instrument and phase anticipation. We define anticipation result of the instrument/phase as the remaining time until the occurrence of one of K surgical instruments or M surgical phases within a future horizon of h minutes. Given a frame index i from video x and an instrument/phase (τ/α), the ground truth for the regression task is defined as:

$$r_h(i, \tau/\alpha) = min\{t_i(\tau/\alpha), h\} \tag{3.1}$$

where $t_i(\tau/\alpha)$ is the remaining time in minutes until the occurrence of τ/α. The ground truth for current time instant T_{obs} ranges $[0, h]$, where 0 denotes that the τ/α is currently presenting and h denotes that τ/α will not happen within next h minutes. Encouraged

from [63], we truncate the target value within h minutes because the next instrument/phase shall not be anticipated accurately for arbitrarily long intervals. This design choice encourages the network only to react when the usage of an instrument in the foreseeable future is likely and predict a constant otherwise. Opposed to current definitions for anticipation tasks, we do not assume an imminent action or rely on dense segmentations.

For each timestamp i, we firstly extract semantic map s_i and instrument bounding boxes b_i. At the same time, we obtain the instrument presence signal t_i and phase signal p_i from the recognition models. The details of the model will be illustrated in the following sections. Given the observed sequence

$$r_h(i_{T_{obs}}, \tau/\alpha) = f\{(x_1, s_1, b_1, t_1, p_1), ...(x_{T_{obs}}, s_{T_{obs}}, b_{T_{obs}}, t_{T_{obs}}, p_{T_{obs}})\} \tag{3.2}$$

, where f is the function modeled by our IIA-Net, and the prediction of T_{obs} is defined by taking all previous features into account.

3.1.2 Classification

As shown in Figure. 1.1, the output space consists of three categories. Background frames, where the $r_h(i, \tau/\alpha) = h$. Anticipating frames, where $0 < r_h(i, \tau/\alpha) < h$. Present frames, where $r_h(i, \tau/\alpha) = 0$. It is notable that these categories can serve as a prior information to limit the range of the regression output. Therefore, we follow the [63] to add a classification objective as a regularization term. This needs our model to predict one of the three categories $c_h(i, \tau/\alpha) \in \{anticipating, present, background\}$, which correspond to an instrument/phase appearing within the next h minutes, being present or a background category when neither is the case.

3.2 Network Architecture

Endoscopic surgery video is a spatial-temporal representation. It spatially represents the visual objects in the scene. Temporally, it represents the dynamic information, e.g., actions, along the time axis. For example, the action performed by the objects can be determine by observing multiple frames. Since predicting only on single frame will largely degrade the performance, conventional model mostly utilize the 2D + 1D or 3D scheme to design

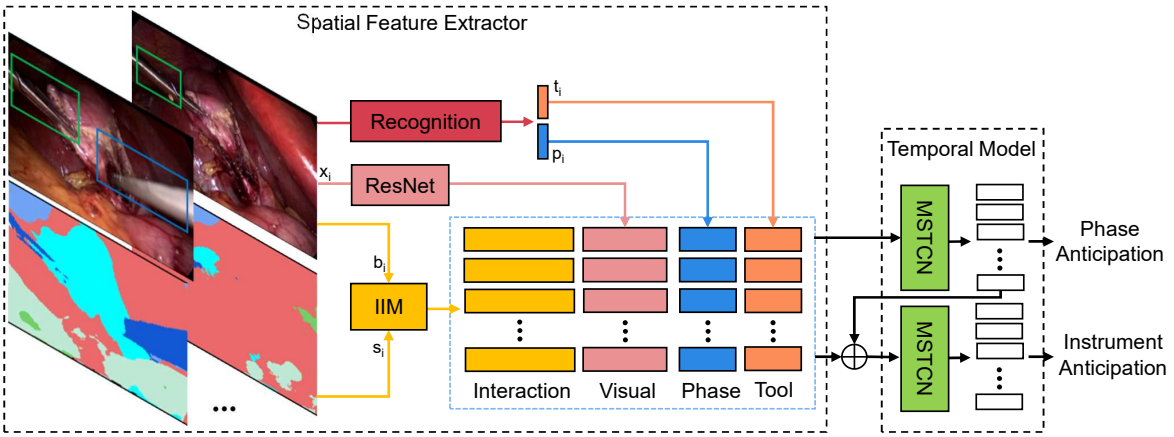


Figure 3.1: Overview of the proposed model. For each frame observed, its estimated semantic map and tool detection are forwarded to instrument interaction module (IIM) to extract interaction feature. The manual annotations for phase and tool signal are fed into temporal model jointly with interaction and visual features.

the network. The 2D + 1D networks apply the 2D convolutional network to extract spatial features and use recurrent neural network or 1D convolutional network to manage the temporal modeling. While the 3D networks mostly utilize the 3D convolution kernels to construct the model. Considering that 3D convolution operation is way more computationally expensive than the 2D + 1D convolution operations, we design our model by dividing the structure into spatial feature extractor for feature extraction and multi-stage temporal convolutional network (MS-TCN) for temporal sequence modeling. According to the running time performance experiment, our network is feasible for real-time use.

3.2.1 Spatial Feature Extractor

Figure. 3.1 shows the overall network architecture of our IIA-Net. It is a two-step model with a feature extractor and a temporal model. The feature extractor combines five inputs x_i, s_i, b_i, t_i, p_i mentioned in Section. 3.1, of which the s_i and b_i are generated from for to model instrument-instrument interactions and the instrument-surrounding interactions. The frame x_i is encoded by ResNet50 [27] into visual features, and the tool signal t_i, phase signal p_i are provided by the recognition model [10] trained upon the Cholec80 dataset. The instrument/phase presence signals are embedded into the feature space and concatenated with interaction feature and visual feature jointly for the input of the next temporal model.

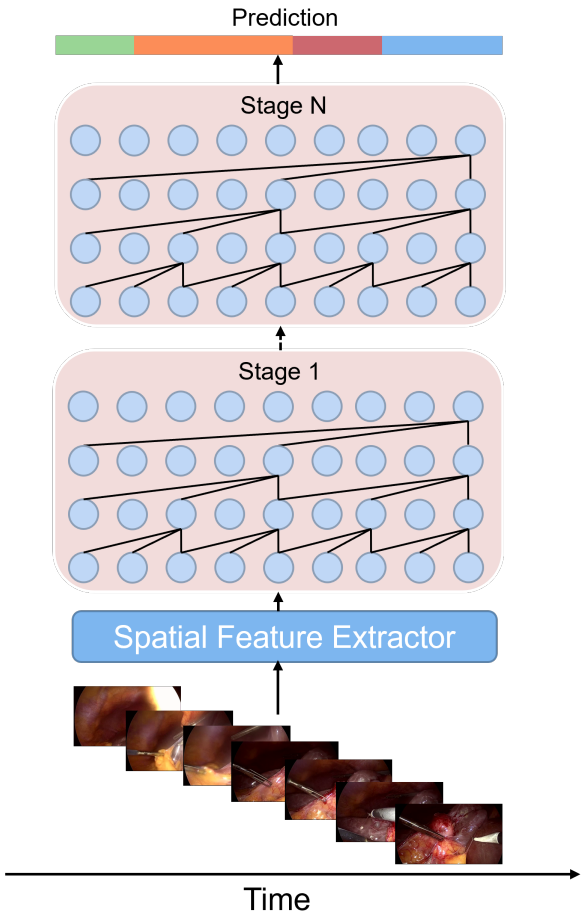


Figure 3.2: Figure about causal MS-TCN. The input video sequence is sequentially fed into the network and the current time's prediction is only conditioned on the previous frames.

3.2.2 Multi-Stage Temporal Convolutional Network

As shown in the Figure. 3.1, for the temporal pattern modeling, we apply a multi-stage temporal convolutional network firstly for phase anticipation. Then we concatenate the above five feature vectors and the prediction of phase anticipation together as the input for the instrument anticipation. In the following section of multi-task learning, we will introduce this strategy in details.

To model temporal patterns in the anticipation task, we modify the MS-TCN [14] to build a lightweight temporal network. The network is constructed fully with dilated temporal convolutions without neither pooling layers nor fully connected layers. Therefore, the model does not suffer from the information loss problem due to the pooling layer, while being able

to process the full resolution temporal sequence. Also, the removal of the fully connected layer reduces the number of the parameters. Compared to the markov chain, the multi-stage temporal convolutional network does not need the prior phases transition information. Therefore, it is more robust to different surgical processes.

To apply our model in an online mode, we use causal convolutions in the network. Instead of the acausal convolutions in [14] with predictions depend on both n past and n future frames, the causal convolutions ensure the prediction of current instant not relying on any n future frames but only depends on the current and previous frames. Specifically, as shown in the 3.2, we pad the input of the last layer and shift remove the output of the following layer. This strategy allows us to easily alter the network to make each timestamp's prediction only conditioned on the previous timestamps.

3.2.3 Instrument Interaction Module

The surgical instruments are the most important representative of the surgeon's mind and they can help to understand the action and intention of the surgeons. Therefore, we capture the instrument's interactions by designing a Instrument Interaction Module (IIM). It consists of two branches, instrument-instrument and instrument-surrounding to indicate surgeon's action and patient's status. In this module, we model surgeons' intention by analyzing the instrument-instrument interaction and instrument-surrounding interaction, shown in Figure. 3.3. We assume each frame is processed to obtain the spatial coordinates and bounding boxes of all instruments. Also, we extract the semantic prior for each frame, which characterizes the semantic class of a region in an image. (e.g., liver, gallbladder).

Instrument-Instrument Encoder This encoder explicitly models the geometric relation among instruments. Here we only consider the interaction between grasper and other instruments because the grasper is the most frequently used instrument. It provides the primary support for the other instruments during the surgery.

We encode the geometric relation $G \in \mathbb{R}^{M\times 4}$ using Eq. 3.3 that is proven effective in object detection [47]. Specifically, at any time instant, given the bounding box of grasper (x_g, y_g, w_g, h_g) and M other instruments in the scene $\{(x_m, y_m, w_m, h_m)|m \in [1, M]\}$, we

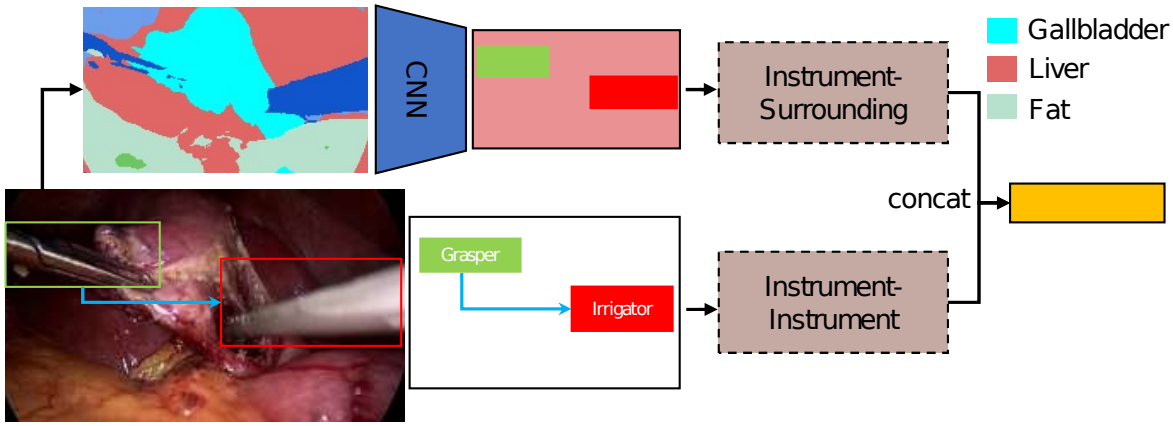


Figure 3.3: Instrument interaction module. Upper: instrument-surrounding modeling uses pooled scene semantic features to encode features; Bottom: instrument-instrument modeling extracts the spatial relations between the grasper and the other instruments.

encode the geometric relation into $G \in \mathbb{R}^{M\times 4}$, the m-th row of which equals to:

$$G_m = [log(\frac{|x_g - x_m|}{w_g}), log(\frac{|y_g - y_m|}{h_g}), log(\frac{w_m}{w_g}), log(\frac{h_m}{h_g})] \tag{3.3}$$

This encoding computes the geometric relation in terms of the geometric distance and the fraction box size. We then embed this geometric feature at each time instant into $\mathbb{R}^{T_{obs}\times C_1}$ where C_1 is the embedding size.

To deal with a more complex interaction that involves more than two surgical instruments, a longer geometric relation vector G is needed. For example, if three instruments are included, a vector with length 12 should be calculated to represent the interaction among the instruments.

Instrument-Surrounding Encoder To encode an instrument's nearby anatomical surroundings, we first extract pixel-level scene semantic classes for each frame. Here, we use a total of $N_s = 7$ scene classes (i.e., background, liver, fat, abdominal wall, tool Shaft, tool tip, gallbladder). Then we transform the integer semantic map into N_s binary masks of the size $T_{obs} \times h \times w$, where h, w are spatial resolution. We apply two convolutional layers on the binary masks with a stride of 2 to get the scene CNN features. We then average the scene feature along the spatial dimensions and generate a feature vector as the encoder's output.

The generated feature vector is in $\mathbb{R}^{T_{obs}\times C_2}$, where C_2 is the number of channels in the convolution layers. After combining the feature vectors from instrument-instrument encoder and

instrument-surrounding encoder, the final feature vector outputed from IIM is in $\mathbb{R}^{T_{obs} \times (C_1+C_2)}$

3.2.4 Detection Model

In this work, the instrument-instrument and instrument-surrounding relation extraction relies on the instrument geometric information to calculate. These requirements inspire us to utilize an object detection model, performing on each frame real time to fully automate the anticipation system.

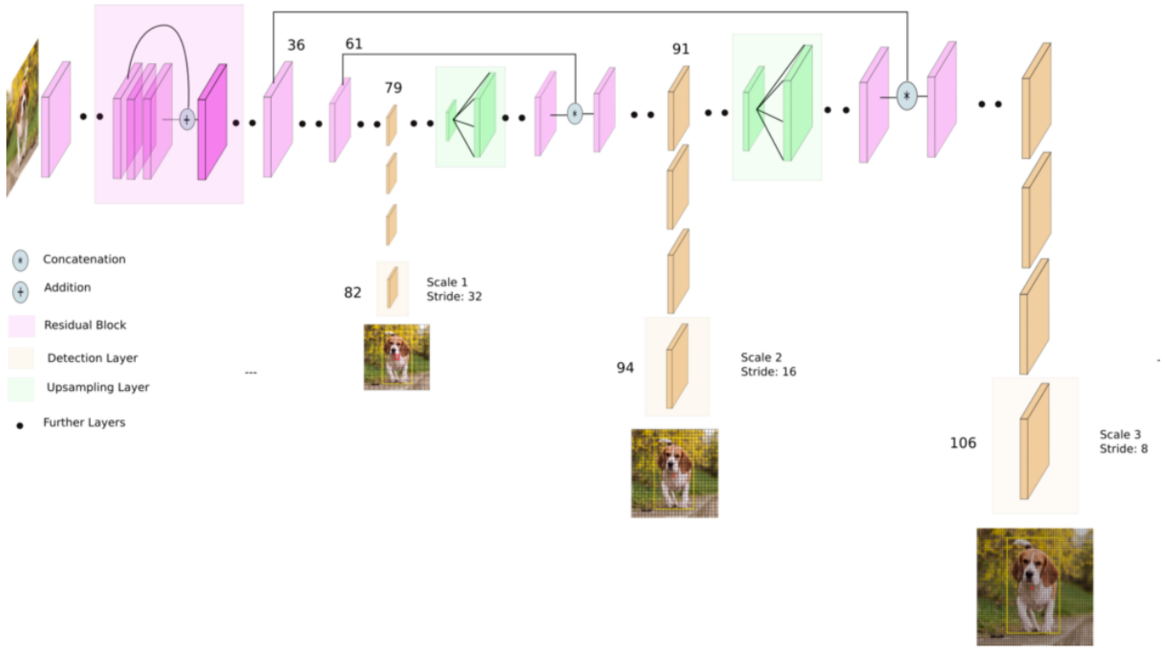


Figure 3.4: Details about the Detection Model. Figure reprinted from [4]

In the conventional computer vision field, there are two main categories of frameworks, one-stage model [15] and two-stage model [61]. The one-stage model directly takes the RGB image as input and output the bounding boxes while the two-stage model needs an object proposal stage. Specifically for the first stage, it proposes all possible objects, and the second stage will determine the most confident ones from those. Straightforwardly, the two-stage model is more time-consuming. In the real-world surgical scenario, the system needs to efficiently perform the decision making without and latency. Therefore, we opt for the one-stage model YOLOv3 [15] for this work.

YOLOv3 is a fully convolutional network, whose output size is controlled by adjusting the convolution step size. Therefore, there is no special restriction on the input image size. Yolov3 draws on the idea of pyramid feature maps. Small-size feature maps are used to detect large-size objects, and large-size feature maps are used to detect small-size objects. As shown in Figure. 3.4, Yolov3 outputs a total of 3 feature maps, the first feature map is down-sampled 32 times, the second feature map is down-sampled 16 times, and the third is down-sampled 8 times.

YOLOv3's entire network has absorbed the essence of Resnet, Densenet, and FPN, and it can be said that it integrates all the most effective techniques for target detection in the current industry. Also, it is extremely fast, more than 1000x faster than R-CNN and 100x faster than Fast R-CNN.

3.2.5 Segmentation Model

In the instrument-surroudning branch, the segmentation map needs to be extracted to provide environmental information. Therefore, we utilize a segmentation model U-Net [64], performing on each frame real time to suppport the IIM. As shown in the Figure. 3.5, the UNet network structure is symmetrical, resembling the English letter U, so it is called UNet. The network is composed of the encoding stage and decoding stage, shown at the left and right, respectively. During the encoder stage, the feature map is extracted and the dimensionality of it is reduced using the maxpooling operation. Opposed to the encoding stage, the decoding stage utilizes the transpose convolution to enlarge the size of the feature map to make final output segmentation map has the same dimensionality as the input image. The grey arrows represent skip-connections, which are used for feature fusion between encoder's feature and decoder's feature. This allows the network to restore the information that is lost during the maxpooling operation.

The U-Net possesses multiple skip connections, allowing the context information propagation from encoding to the decoding feature maps. Also, the larger number of feature channels make it easy to propagate the information to the large-resolution layers. Therefore, the expansive path is more or less symmetric to the contracting part, and yields a u-shaped

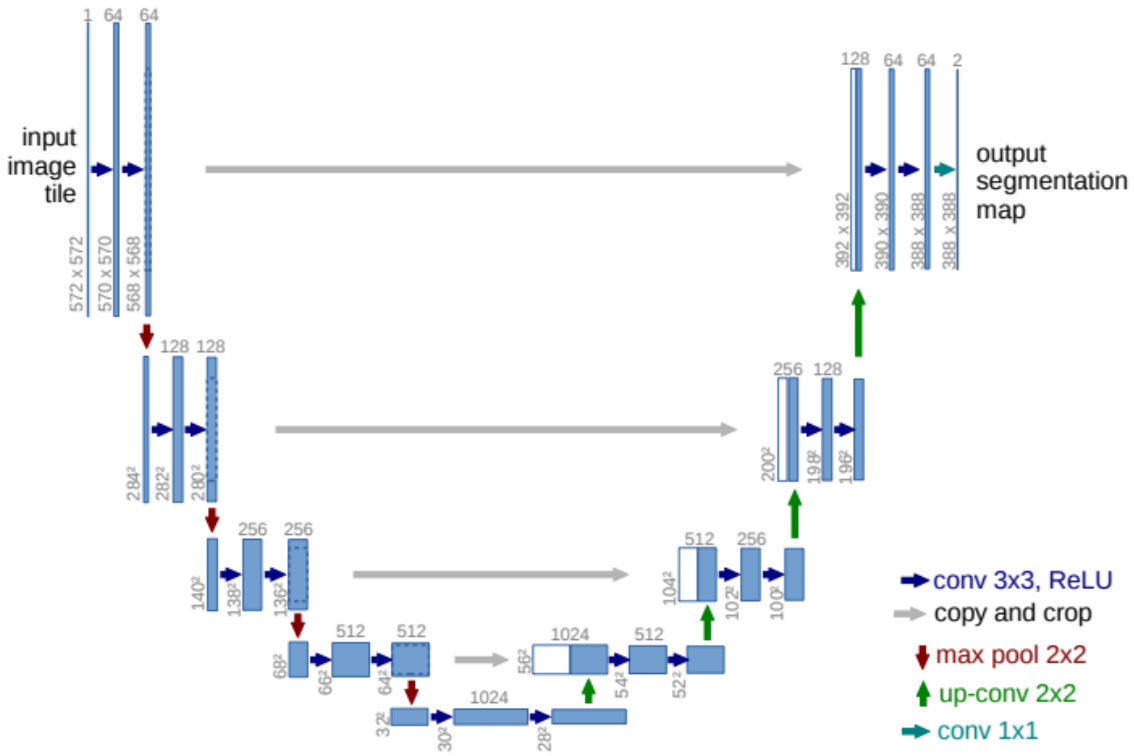


Figure 3.5: Details about the UNet. During the encoding, the size of the feature map is reduced and the hidden dimension is enlarged. In the decoding stage, the size of feature map is enlarged. Figure reprinted from [64]

architecture. The U-Net is a fully convolutional network, which only uses the valid part of each convolution without any fully connected layers. Also, the U-Net is feasible to handle the large images in real time, since otherwise the resolution would be limited by the GPU memory in the other segmentation models.

3.2.6 Recognition Model

In this work, not only the tool detection feature is used in our work, the phase presence signals are also important prior to indicate the current occurrence information of the surgical instrument/phase. Acccording to the previous papers, prior works on instrument/phase detection on Cholec80 is reasonably accurate, meaning that we can transfer the pretrained recognition model to our system without any obstacle. Considering the running time performance for the real-world surgical scenario, we opt for the TeCNO [10] to process the endoscopic video

frame by frame and provide the presence signal. Also, our experiment upon running time shows that this recognition model is capable of real-time task. In the newest experiment, we train the network using the signals from the recognition models, and the experimental results shown in this thesis is based on the predicted phase/tool presence signals instead of the ground truth.

3.3 Loss Functionality

We train our model with a combination loss: the **regression loss** which forces the model learn the accurate value for remaining time prediction, and **classification regularization loss** to provide another perspective to process the anticipation. The classification loss encourages the model to determine the ending trend which is helpful for the entire task. For the regression loss, we follow the [63] to apply smooth L1 loss because of its benefit for stable training convergence. For the classification loss, we apply cross-entropy to estimate the difference between the prediction and ground truth.

3.3.1 Smooth L1 Loss

Multiple losses, e.g., L1, L2 and smooth L1 loss are widely used for the regression task.

$$SmoothL1 = \begin{cases} 0.5(f(x) - y)^2 & if\ |f(x) - y| \langle\ 0.5 \\ |f(x) - y| - 0.5 & otherwise \end{cases} \tag{3.4}$$

In this work, we opt for the smooth L1 loss for a faster and stable convergence, shown in Eq. 3.4. Compared to the L1 loss, L2 loss is better because L2 loss is differentiable everywhere, and the gradient value is also dynamically changing, which can quickly converge; while L1 loss is not differentiate at 0 point, and its gradient remains unchanged. For a small loss value, the gradient is very large. In deep learning, it is necessary to use a varying learning rate to reduce itself when the loss value is very small. However, the L1 loss can better handle the outliers (exceptions) compared to the L2 loss. In this work, the outliers does not need to be detected. Therefore, we choose L1-based loss function to be less sensitive to outliers and more stable for the training procedure.

The difference between the Smooth L1 and L1 Loss functions is that the derivative of L1

Loss at 0 is not differentiate, affecting the convergence. However, the smooth L1 loss adds square function at the 0 point to make it smoother to solve this problem. Compared with the L1 loss function, the Smooth L1 Loss can converge faster. Compared with the L2 Loss function, it is insensitive to the outliers, and the gradient change is relatively smaller, making it difficult to collapse during training.

3.3.2 Cross Entropy Loss

Cross entropy describes the distance between the distribution of prediction and ground truth, when the cross entropy is smaller, the closer the two distribution are.

$$CrossEntropy = -1/N \sum_{i} \sum_{c=1}^{M} y_{ic} log p_{ic} \tag{3.5}$$

, where the y is the ground truth probability value of the t^th sample and c^th category, and the p is the predicted value. Although cross entropy describes the distance between two probability distributions, the output of a neural network is not necessarily a probability distribution. For this reason, we often use Softmax function to turn the results of neuron obtained by the forward propagation of the neural network into a probability distribution.

Softmax is often used in the multi-classification process. It normalizes the output of multiple neurons to the $(0, 1)$ interval, so the output of Softmax can be regarded as a probability for multi-classification.

$$Softmax = \sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \quad for\ i = 1, 2, \ldots, \tag{3.6}$$

3.4 Multi-Task Learning

As the instrument presence has a strong correlation with the phase presence, we perform a multi-task learning strategy to jointly handle instrument and phase anticipation at the same time. This help us to reduce the model parameter by merge two task into one model instead of assigning each task a model. Also, this is computational feasible for the future deployment to the hospital's edge devices, because the large central data server has many ethical and privacy issue.

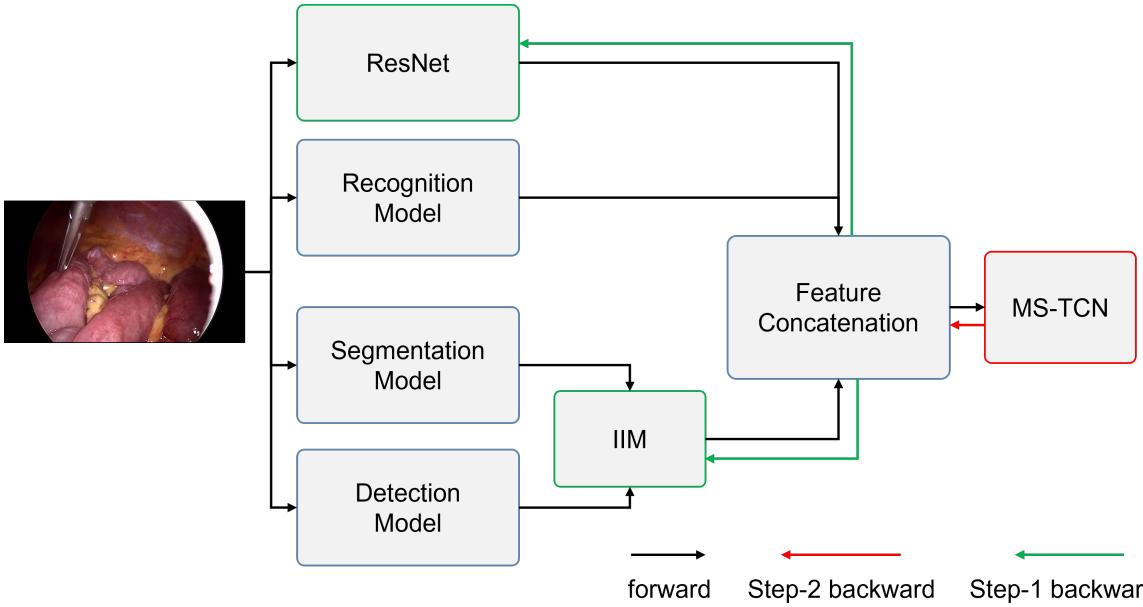


Figure 3.6: Gradient flow. The green lines and boxes are optimized on the first step of training. The red line and box is trained on the second step. The other models are not optimized during the traning process.

Based on the experimental experience, we find that the phase pattern of surgical workflow is less variable and is consecutive. Therefore, the performance of phase anticipation is better than the instrument. To alleviate the problem of instrument anticipation, we firstly predict the phase anticipation result based on the mentioned network, as shown in Figure. 3.1 Then the phase anticipation result is concatenated with the features for the instrument anticipation. This strategy can help us to propagate the information from phase to the instrument in an weakly supervised manner.

3.5 Gradient Flow

Limited by the computational resource, fully end to end training from raw RGB frame is not feasible. Therefore, we opt for the two-step training strategy, and stop the gradient back propagation when it comes to optimize the detection model, segmentation model, and recognition model. This operation makes the peripheral models not trainable when we optimize the main network. Because the peripheral models are already pretrained and evaluated on the other dataset, the provided tool detection signals are faithfully and can be

relied on. Also, experiments show that the anticipation system is robust to the noisy tool detection signals. As shown in the Figure. 3.6, the parameters of three models are frozen during the training. For first step of training, we train the ResNet with the IIM module without any temporal modeling method, as shown in the red line in Figure. 3.6. Then, for the second step of training, we freeze all four models, including ResNet and IIM to only optimize the temporal modeling method, MS-TCN.

Chapter 4

EXPERIMENT

4.1 Datasets and Preprocessing

4.1.1 Anticipation Dataset

The size of dataset matters, especially for the deep learning methods to train a robust model for different scenarios. In this work, we leave 20 video samples from the dataset for the evaluation. Also, there is not training on the evaluation samples, avoiding the data leaking problem. Also, we conduct the two-step training strategy and the normalization layer is not used for the seuquence modeling, meaning that the model is truly causal. We do not consider different size of training set to optimize the model because the goal of this thesis is to propose a powerful framework for the regression-based anticipation task, not the few or zero shot learning. We need a large dataset with endoscopic surgery videos to run the empirical analysis of this surgical workflow anticipation method. The instrument/phase presence annotation should be provided in the dataset to produce a precise estimation of the anticipation signal. Specifically, the dataset should contain the frame-wise annotations that indicate which instrument is currently used and which phase is currently going on. To largely collect dataset from the real world needs a lot of regulation and ethical concerns, and we consider it the next step to follow in future work.

We evaluate our method on publicly available surgical workflow intra-operative video dataset, Cholec80 [73], which contains laparoscopic cholecystectomy procedures for the resection of the gallbladder. In the Cholec80 dataset, it is proposed to cope with the limitation that the m2cai16-workflow does not contain tool binary annotations. Therefore, it contains both phase and tool binary annotations. It contains cholecystectomy surgical videos performed at the University Hospital of Strasbourg in France, where each surgical video frame is labeled with binary annotations indicating tool/phase presence. Cholec80 dataset consists of 80 videos ranging from 15 minutes to 90 minutes. We follow the same split as [63], separating the

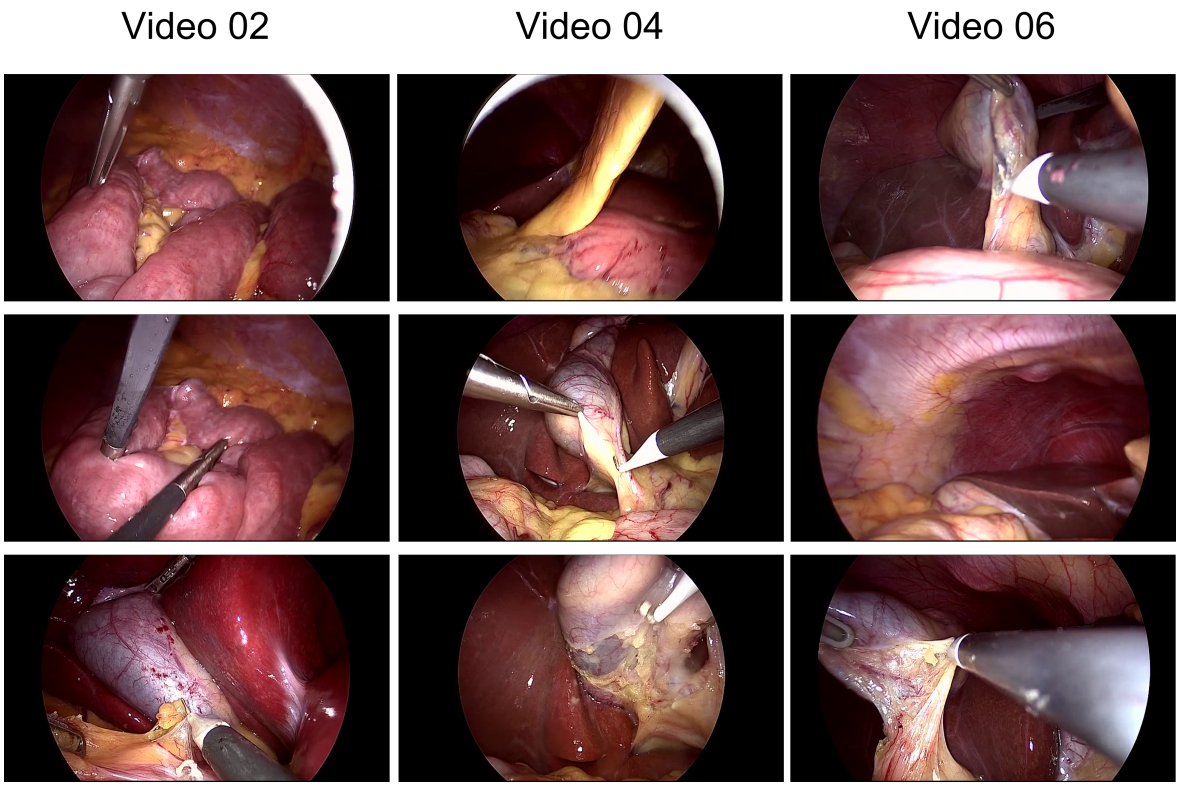
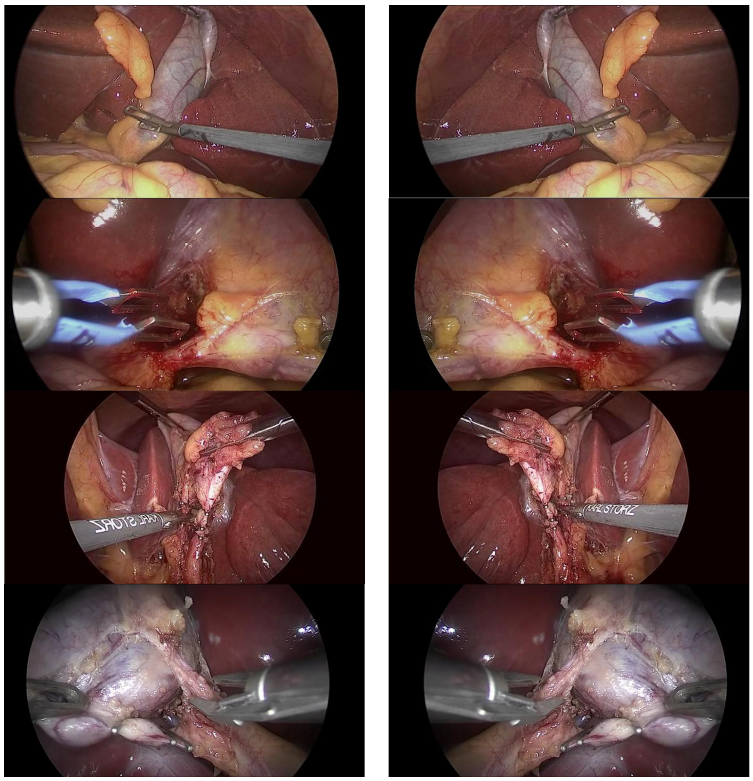


Figure 4.1: Samples from Cholec80.

dataset to 60 videos for training and 20 for testing. We resize the video's spatial resolution from 1920×1080 to 224×224 to reduce the computational cost dramatically. This is because the detection and segmentation performance are not largely affected by the resolution of the frame. Also, we resample the video from 25 fps to 1 fps. Specifically, we select the first frame for each second.

4.1.2 Detection Dataset

As mentioned above, we need to extract instrument bounding boxes for the IIM module to produce instrument-instrument information. However, there is no fine-grained bounding box annotation in the Cholec80 dataset, and it is unfeasible for us to annotate it manually. Therefore, we follow the transfer learning widely used to handle the no ground truth problem in the computer vision field. Specifically, we need a detection dataset to train the detection model. Then, we freeze the parameters of the detection model and utilize it to perform inference upon the Cholec80 dataset.




```
<annotation>
        <folder>VOC2007</folder>
        <filename>v01_025150.jpg</filename>
        <source>
                <database>0</database>
                <annotation>0</annotation>
                <image>0</image>
                <flickrid>0</flickrid>
        </source>
        <owner>
                <flickrid>0</flickrid>
                <name>0</name>
        </owner>
        <size>
                <width>596</width>
                <height>334</height>
                <depth>3</depth>
        </size>
        <segmented>0</segmented>
        <object>
                <name>Clipper</name>
                <pose>Unspecified</pose>
                <truncated>0</truncated>
                <difficult>0</difficult>
                <bndbox>
                        <xmin>142</xmin>
                        <ymin>214</ymin>
                        <xmax>268</xmax>
                        <ymax>322</ymax>
                </bndbox>
        </object>
        <object>
                <name>Grasper</name>
                <pose>Unspecified</pose>
                <truncated>0</truncated>
                <difficult>0</difficult>
                <bndbox>
                        <xmin>260</xmin>
                        <ymin>11</ymin>
                        <xmax>370</xmax>
                        <ymax>72</ymax>
                </bndbox>
        </object>
</annotation>
```


Figure 4.2: Samples from detection dataset [33]

Here we utilize the dataset proposed in [33]. Among the previous works, the are few datasets supporting automated tool identification. Most of them are about frame-level tool presence detection. The m2cai16-tool from M2CAI 2016 Tool Presence Detection Challenge and Cholec80 both miss the bounding box information. Then, the [33] expands upon the m2cai16-tool and introduces a new dataset to study the task of surgical tool localization in real-world laparoscopic surgeries and to enable higher-level analysis of surgical videos. The dataset contains annotations of spatial bounds of tools, named m2cai16-tool-locations. It consists of 15 videos of cholecystectomy procedures. The videos are ranging from 20 to 75 minutes and are downsampled to 1 fps. The annotated tools are grasper, bipolar, hook, scissors, clip applier, irrigator, and specimen bag. The labeling process of [33] is supervised by the spot-checking from a surgeon, with the coordinates of spatial bounding boxes around the tools.

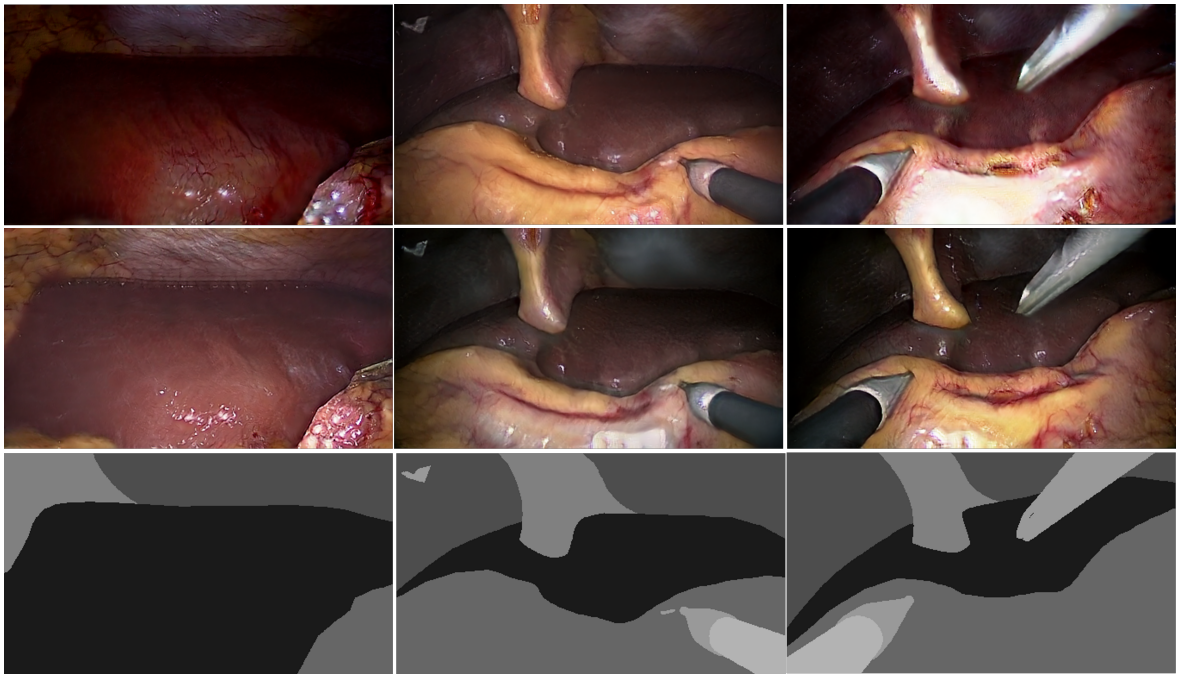


Figure 4.3: Samples from segmentation dataset [33]

4.1.3 Segmentation Dataset

As mentioned above, we also need to extract segmentation map for each frame to provide instrument-surrounding information for the IIM. Therefore, we follow the same transfer learning strategy to firstly train a segmentation model on segmentation dataset and then transfer the model upon the Cholec80 to perform the inference.

It is manually unfeasible to annotate semantic map frame by frame. Therefore, there is not such dataset that considers this segmentation map estimation task in the medical domain. In this work, we utilize the dataset proposed in [58] to train our segmentation model. [58] is a project that translates medical images among different modalities. Specifically, it generates synthetic images in a 3D environment to roughly represent the laparoscopic liver surgery scenes. Then it utilizes the generative adversarial network (GAN) to discriminate if the image is from the real surgery or simulated domain. This forces the generated images to look like real laparoscopic images. As shown in the Figure. 4.3, we use the translated images along with their semantic labels as the training data for the segmentation model.

4.2 Setups

4.2.1 Hardware and Software Framework

We use the two different setups to accomplish our whole project.

- The first setup has 7 core CPU with 12 GB of memory facilitated with two Nvidia RTX 2080ti GPUs.
- The second setup has 7 core CPU with 24 GB memory equipped with an Nvidia RTX 3090 GPU.

The first setup is built upon the linux ubuntu 16.04 system and the second setup is built upon the windows 10 operating system. We used python programming language and several deep learning libraries of python. To build up the deep learning environment for the project, we utilize the Pytorch 1.3.0, Pytorch-lightning and cudatoolkit 10.2. Furthermore, for data pre-processing and dataset making, we utilize torchvision and OpenCV. The visualization of jointly video and prediction is based on the matplotlib.

4.2.2 Training Setting for YOLOv3

For the training of detection model, we adopt the codebase from https://github.com/ultralytics/yolov3 to train the YOLOv3 upon the detection dataset. We firstly convert the localization annotation into VOC format and then set up the data's file path for the training code.

The learning rate is initialized as 0.001 and decay every 3800 training steps. Since the weights of a neural network are updated based on the minibatch, it is possible that the samples can not represent well inside the minibatch and lead to sharp change of weights during the training. Therefore, the parameter momentum is applied to prevent this situation. The momentum is set as 0.9 to fluctuate the weight update a bit and penalize large weight changes between iterations.

Also, we applied the weight decay parameter as 0.0005 to prevent the overfitting during the training. The neural network contains millions of weights and can easily overfit to the training data. This leads the model perform well on the training set but degrade on the test set. This

is because part of the neurons are dominating the model and one of the ways to mitigate this problem is to penalize large value for weights. Therefore, we apply a regularization term as weight decay parameter.

During the training, the conventional data augmentation is applied to make the model robust to different scenarios. The flipping, cropping and translation are used in our case.

4.2.3 Training Setting for UNet

For the segmentation map, we train the UNet with the following training settings upon the segmentation dataset. The network is trained for 400 epochs with a batch size of 32. We employ the Adam optimization function with a learning rate $1e-5$ for the training. Moreover, momentum parameters are $\beta_1 = 0.9$ and $\beta_2 = 0.999$ same as default and epsilon is none. Moreover, for the training of U- Net, we use the cross entropy loss function.

We also apply the early-stopping function by inspecting the trend of validation loss. If the validation loss does not decrease for 5 epochs or even increase, we will stop the training process and run the test process. If it keeps decreasing, then the training process will continue until the end or the validation increase. This strategy greatly avoid the overfitting problem and can shorten the training process.

During the training, the learning rate is altered accordingly. There are multiple learning rate decay strategies, but most of them are based on the experienced tuning and highly depend on the task. Here we utilize the ReduceLROnPlateau, monitoring the learning rate during the training period and modify it depending on the validation loss. When the validation loss is not decreasing for several epochs, this strategy will decrease the learning rate by $\sqrt{0.1}$ and the the new learning rate is:

$$lr_{new} = lr_{old} * \sqrt{0.1} \tag{4.1}$$

Also, the lower bound is set up as $0.5e-6$ to keep the learning rate from being extremely low.

4.2.4 Training Setting for IIA-Net

As mentioned in the previous sections, we adopt a two-step training strategy and here are the details. The whole training setting can be divided into two parts, spatial settings and temporal settings.

- Spatial settings. In the spatial training, we mainly train the ResNet-based feature extractor. The ResNet is pretrained on the ImageNet [43], and we remove the last few layers and add multiple perceptron layers to support the multi-task learning. The early stopping strategy is based upon the validation loss of instrument anticipation result. The batch size is set as 80 and the frame rate is downsampled into 1. The model is at least trained for 40 epochs and at most for 100 epochs.

- Temporal settings. In the temporal training, we only train the MS-TCN temporal model, MS-TCN. Because of the sequence modeling characteristic, the batchsize is set as 1. The learning rate is set as 0.002 and the whole training process will not exceed 100 epochs. The least training epochs is 30.

4.2.5 Adam Optimizer

Note that all three models mentioned above are optimized by the Adam optimization function [40]. Adam algorithm is an algorithm that performs the first-order gradient optimization on a random objective function. It is easy to implement, and has high computational efficiency and low memory requirements. The diagonal rescaling of the Adam algorithm gradient is invariant, so it is very suitable for solving problems with large-scale data or parameters.The algorithm is also suitable for solving non-stationary problems with large noise and sparse gradients. The empirical results also show that the Adam algorithm is comparable to other stochastic optimization methods in practice.

4.3 Evaluation Metrics

4.3.1 Quantitative Metrics

Automatic instrument preparation is one of the primary tasks that benefits from surgical workflow anticipation. In this work, it outputs the specific value to indicate the remaining time of the next tool's/phase's occurrence. Therefore, we follow [63] to opt for the frame-based evaluation metrics, mean absolute error (MAE) and its variants, i.e., inMAE, pMAE and eMAE. The inMAE, eMAE, pMAE can be represented in the following formulas:

$$inMAE = \frac{1}{T}\sum_{i}^{T} MAE(p, r(x_i, \tau/\alpha)), 0 < r(x_i, \tau/\alpha) < h \tag{4.2}$$

$$eMAE = \frac{1}{T}\sum_{i}^{T} MAE(p, r(x_i, \tau/\alpha)), 0 < r(x_i, \tau/\alpha) < 0.1h \tag{4.3}$$

$$pMAE = \frac{1}{T}\sum_{i}^{T} MAE(p, r(x_i, \tau/\alpha)), 0.1h < p < 0.9h \tag{4.4}$$

, where the p is the prediction of the model and $r(x_i, \tau/\alpha)$ represents the ground truth value for the current timestamp. In specific, we average the MAE of 'anticipating' frames using inMAE, because the preparation system should only react to the signals that indicate tool/phase is anticipating. Also, it does not require tools or phases to be anticipated too far in advance. Therefore, we propose to use eMAE to evaluate intervals ($0 < r(x_{T_{obs}}, \tau/\alpha) < 0.1h$) that provides the most effective support to the computer-assistance system.

Following [63], we utilize the pMAE to measure the precision performance of our model. As instrument's/phase's occurrence is sparse, it is not always predictable. Therefore, a low recall does not necessarily indicate poor performance, making precision metrics popular for anticipation [21]. Similarly, the idea of precision in pMAE adopts the same concept of precision in classification task. We then apply the pMAE as the MAE of predictions with $0.1h < f(x) < 0.9h$.

4.3.2 Sample Results

The surgical workflow anticipation from endoscopic surgery video is not a fully defined task, and there are only a few research papers were published based on these chores. Also, those works do not share the same task format, and the outputs are various. For example, as mentioned in the introduction, current works output a short sequence of the future. Each timestamp of the sequence is predicted with pre-defined labels, indicating the occurrence of a specific instrument/phase occurrence. Therefore, we randomly select a few samples and visualize them using the formulation in this work to straightforwardly show the anticipation signal for the surgeons. Also, we plot the segmentation map and detection results to show the performance of segmentation model and detection model, respectively.

Anticipation. This work adopts a different way to handle the anticipation task by predicting the remaining time of the next occurrence. Therefore, we plot a line chart to show the difference between the prediction and ground truth along the time axis. Also, it is an instrument/phase-wise chart capable of showing the performance in the multi-label scenario, where multiple instruments could present simultaneously.

Detection Segmentation. In this work, we extract the instruments detection and segmentation map for each frame in advance. However, as mentioned above, we adopt transfer learning to perform inference upon the Cholec 80 dataset because there is no ground truth label for detection and segmentation on Cholec80. However, the performance of such models on Cholec 80 is unknown. Therefore, we randomly select multiple frames from different videos and plot the segmentation map from our trained UNet. For the detection evaluation, we adopt the same strategy to evaluate the performance of detection upon pre-constructed test set by [33]. These largely ensure the quality of features that we will use for the capturing instrument-instrument and instrument-surrounding relations.

4.3.3 Earliest Consistency Criteria

In this work, the system outputs a regression value to show the remaining time until the occurrence. However, this regression approach has a limitation, which is the system can not forecast the occurrence at very early stage. Therefore, the result of the anticipation is

usually better when it is close to occurrence compared to the timestamp that is far from the occurrence. This result can be found by investigating the eMAE. Therefore, to show how early we can predict the future with an acceptable error tolerance, we propose the ea criteria to set up an error threshold and find the earliest prediction that has smaller error than the threshold. The criteria is shown in the following:

$$ea@a = min(i), |p_i - r(x_i, \tau/\alpha)| < a \tag{4.5}$$

, where i denotes the timestamp, the a denotes the error threshold, p denotes the prediction and $r(x_i, \tau/\alpha)$ is the ground truth. We visualize predictions based on this criteria.

Chapter 5

RESULTS AND DISCUSSIONS

In this section, we quantitatively and qualitatively demonstrate the result of anticipation system. Specifically, the instrument/phase-wise results are given. Also, the corresponding discussions are shown and explained in detail for each experimental factor. Furthermore, the ablation study is conducted to show the effect of different model architecture settings, including the number of stages in MS-TCN and the introduction of IIM. In the end, the correlation between instruments and phases is shown and extracted segmentation map is plotted. The results of the baseline methods [63] are reproduced by retraining the public model and evaluating on our metrics.

5.1 Anticipation Results

5.1.1 Quantitative Results

We firstly evaluate the MAE performance with respect to the previous mentioned quantitative metrics. Specifically, the inMAE, pMAE and eMAE are used for instrument and phase anticipation, respectively. We evaluate the inMAE for the 'anticipating' frames, and eMAE for the frames that provide the most effective support for the robotic system. Also, the pMAE

Table 5.1: inMAE comparison. We report the mean over instrument types in minutes per metric. Ours 2D: our feature extractor without temporal training.

	Instrument		
Model	h = 2min	h = 3min	h = 5min
MeanHist	1.09	1.62	2.64
OracleHist (offline)	(0.92)	(1.31)	(2.01)
Baseline [63]	0.77	1.13	1.80
Ours 2D	0.70	1.07	1.65
Ours	**0.66**	**0.97**	**1.48**

Table 5.2: pMAE comparison. We report the mean over instrument types in minutes per metric. Ours 2D: our feature extractor without temporal training.

	Instrument		
Model	h = 2min	h = 3min	h = 5min
MeanHist	0.93	1.34	2.14
OracleHist (offline)	(0.83)	(1.18)	(1.73)
Baseline [63]	0.64	0.92	1.49
Ours 2D	0.52	0.77	**1.16**
Ours	**0.42**	**0.69**	1.28

is utilized to measure the precision concept in the regression task. We remove the horizon setting of 7 minutes since anticipating the surgical workflow too early is unnecessary for instrument preparation and robot assistance. We re-implement methods from [63] and retrain them as the baseline methods for phase anticipation. We evaluate the model for instrument and phase anticipation on horizons of 2, 3, and 5 minutes. For the phase anticipation, the performance of this work is generally better than the baseline. The MeanHist and OracleHist methods are two histogram-based baseline methods to anticipate the future. For MeanHist, segments are expanded to the mean video duration. For OracleHist, we expand the segments to the real video duration at train and test time [63].

As shown in the Table. 5.1 and Table. 5.2, our IIA-Net achieves lower inMAE and pMAE error compared to the previous methods for the instrument anticipation. Regarding the pMAE error, the margin increases even further. Even though [63] is trained in an end-to-end fashion, it is also outperformed by our IIA-Net, which is trained in a two-step process. Ideally the MAE, including inMAE and pMAE should be close to 0.3 for the real-world application. This value is theoretical from the experiment, a more precise number should be estimated by passing different systems to the surgeons to determine.

Interestingly, our model trained without temporal context (Ours 2D) achieved a lower pMAE when $h = 5$. This is because the 2D model has difficulty foreseeing long-horizon occurrence and easily predicts the value that is close to $0/h$, making its prediction unsmooth. Also, our model achieves the lowest eMAE error, seen from Table. 5.5 and Table. 5.6. This suggests

Table 5.3: inMAE comparison. We report the mean over phase types in minutes per metric. Ours 2D: our feature extractor without temporal training.

	Phase		
Model	h = 2min	h = 3min	h = 5min
Baseline [63]	0.63	0.86	1.17
Ours 2D	0.70	1.04	1.40
Ours	**0.62**	**0.81**	**1.08**

Table 5.4: pMAE comparison. We report the mean over phase types in minutes per metric. Ours 2D: our feature extractor without temporal training.

	Phase		
Model	h = 2min	h = 3min	h = 5min
Baseline [63]	0.62	0.85	1.37
Ours 2D	0.53	0.76	**1.12**
Ours	**0.49**	**0.73**	1.22

that our model can effectively identify instrument or phase occurrence a few seconds ahead. In real-world scenarios, this is typically the most critical time for accurate anticipation.

For the eMAE metric in Table. 5.5 and Table. 5.6, our work achieves the best performance with a large improvement, especially when the horizon is 5 min. The larger the horizon is, the bigger improvement we will achieve. This suggests that our work is sensitive about the triggering activities that are near to the occurrence. Because, larger horizon allocates the model a larger view when anticipating the future. And it will introduce more uncertainties because the model has a larger interval to consider. Therefore, the model could be easily affected by this because it might learn the pattern of the start of 'anticipating' frames. This pattern is not discriminative and is not robust enough to be applied to end of 'anticipating' frames (eMAE). Therefore, a smaller eMAE in a larger horizon indicates that the system can learn the pattern from the frames with $0 < r(x_{T_{obs}}, \tau/\alpha) < 0.1h$ that provides the most effective support to the computer-assistance system.

Table 5.5: eMAE comparison. We report the mean over instrument types in minutes.

	Instrument		
Model	h = 2min	h = 3min	h = 5min
MeanHist	1.85	2.72	4.35
OracleHist (offline)	(1.36)	(1.93)	(2.96)
Baseline [63]	1.12	1.65	2.68
Ours 2D	1.07	1.65	2.51
Ours	**1.01**	**1.46**	**2.14**

Table 5.6: eMAE comparison. We report the mean over phase types in minutes.

	Phase		
Model	h = 2min	h = 3min	h = 5min
Baseline [63]	**1.02**	1.47	1.54
Ours 2D	1.38	1.85	2.42
Ours	1.18	**1.42**	**1.09**

5.1.2 Instrument/Phase-wise Result

In this section, we give the detail of anticipation results for each instrument and phase to examine which instrument/phase is the easier one to predict and which instrument/phase is the harder one. Here, the instrument/phase-wise errors are shown in minutes for horizons of 2, 3 and 5 min. Note that we remove result of the bipolar and grasper because they almost always present in the surgery.

As shown in the Table. 5.7, the instrument-wise errors of 5 min is generally larger than the errors of 2 min and 3 min. The larger the horizon is, the bigger the error is. This is because the large horizon allocates a large range of values for the model to regress. Therefore, the difference between prediction and ground truth will grow when the horizon grows.

Having a horizon-agnostic observation, the scissor is the easiest instrument to anticipate and has the lowest MAE value. Compared to the other instruments, the anticipation of scissor is stable with little variance. This is possible because the scissor's occurrence is strongly correlated to the clipping cutting phase, and we introduce the phase presence signal in our

Table 5.7: Instrument-wise performance upon inMAE/eMAE for instrument anticipation task.

	h = 2min				
	Bipolar	Scissors	Clipper	Irrigator	Specimen Bag
Baseline [63]	0.88/1.46	0.55/0.89	0.85/1.37	0.83/1.23	0.64/0.66
Ours 2D	0.62/1.16	0.42/0.81	0.73/1.25	0.84/1.31	0.46/0.55
Ours	0.64/1.13	0.40/0.87	0.64/1.18	0.71/1.06	0.38/0.47
	h = 3min				
	Bipolar	Scissors	Clipper	Irrigator	Specimen Bag
Baseline [63]	1.24/2.02	0.85/1.14	1.31/2.15	1.13/1.66	1.01/1.26
Ours 2D	1.01/1.92	0.80/1.04	1.04/1.78	1.04/1.58	0.99/1.06
Ours	1.00/1.90	0.79/1.09	1.02/1.77	0.96/1.47	0.91/1.08
	h = 5min				
	Bipolar	Scissors	Clipper	Irrigator	Specimen Bag
Baseline [63]	1.96/3.47	1.55/1.89	2.08/3.75	1.81/2.86	1.36/1.41
Ours 2D	1.98/3.21	1.49/1.78	2.07/3.72	2.27/3.07	1.36/1.46
Ours	1.59/3.06	1.38/1.46	1.47/2.85	1.54/2.39	1.44/1.41

model. Also, the scissor usually presents after the clipper finishing the clipping action. Therefore, the scissor can also anticipated by observing the occurrence of the clipper. The similar thing happens to the anticipation of sepcimen bag, which also achieves a better performance compared to the other instruments. When the surgeon finishes the dissection of the gallbladder, the surgery will go into gallbladder retraction phase and the specimen should be introduced to package the gallbladder for the following removal. This is a significant triggering activity and can be well captured by our model.

Among the instruments, the bipolar and irrigator are the hardest ones to anticipate. This is because the they are usually used to stop the bleeding of anatomical structures. However, the introduction of such instruments is not fully depended on the bleeding activity, it owns a degree of randomness due to different surgeons. Because, the surgeons of different skills will process the complication accordingly. This make it hard for system to be aware of when

Table 5.8: Phase-wise performance upon inMAE/eMAE for phase anticipation task.

	h = 2min					
	P2	P3	P4	P5	P6	P7
Baseline [63]	0.45/0.69	0.72/1.36	0.52/0.91	0.68/1.22	0.78/1.37	0.48/0.57
Ours 2D	0.51/1.04	0.85/1.69	0.53/1.08	0.86/1.72	0.64/1.32	0.61/1.46
Ours	0.50/0.88	0.77/0.59	0.45/0.82	0.77/1.42	0.45/0.90	0.59/1.03
	h = 3min					
	P2	P3	P4	P5	P6	P7
Baseline [63]	0.87/1.24	0.86/1.76	0.68/1.88	0.85/1.79	0.84/1.75	0.67/1.39
Ours 2D	0.95/1.81	1.17/2.24	0.77/1.59	0.98/2.12	1.10/1.40	0.99/1.95
Ours	0.57/0.91	0.99/1.92	0.68/1.07	1.00/1.97	0.73/1.13	0.80/1.64
	h = 5min					
	P2	P3	P4	P5	P6	P7
Baseline [63]	0.75/0.96	1.34/2.57	1.09/0.82	1.27/1.83	1.14/1.96	1.11/1.09
Ours 2D	0.76/0.98	1.85/3.99	1.16/0.87	1.70/3.79	1.41/2.21	1.30/2.69
Ours	0.77/0.89	1.04/1.91	1.07/1.13	1.39/0.66	1.08/1.20	1.19/0.98

is the right time to handle bleeding and introduce the bipolar and irrigator. Also, due to the patient's variance and various video quality, the RGB frame can not effectively represent this situation, especially under the limit training data.

As shown in the Table. 5.8, the easiest phase to anticipate is the phase 3, clipping cutting phase. This phase aims to clip and cut the cystic artery and duct when two and only two structures entering the gallbladder which is still attached only by the upper part of the liver bed after the triangle dissection. In this phase, the clipper is firstly introduced followed by the scissor to cut the cystic artery. Benefited from the tool presence signal from recognition model, this work can determine the start of the clipping cutting phase from the occurrence of clipper and scissor. The hardest phase to anticipate is the phase 5, cleaning coagulation, because of the uncertainty mentioned before.

5.1.3 Sample Results

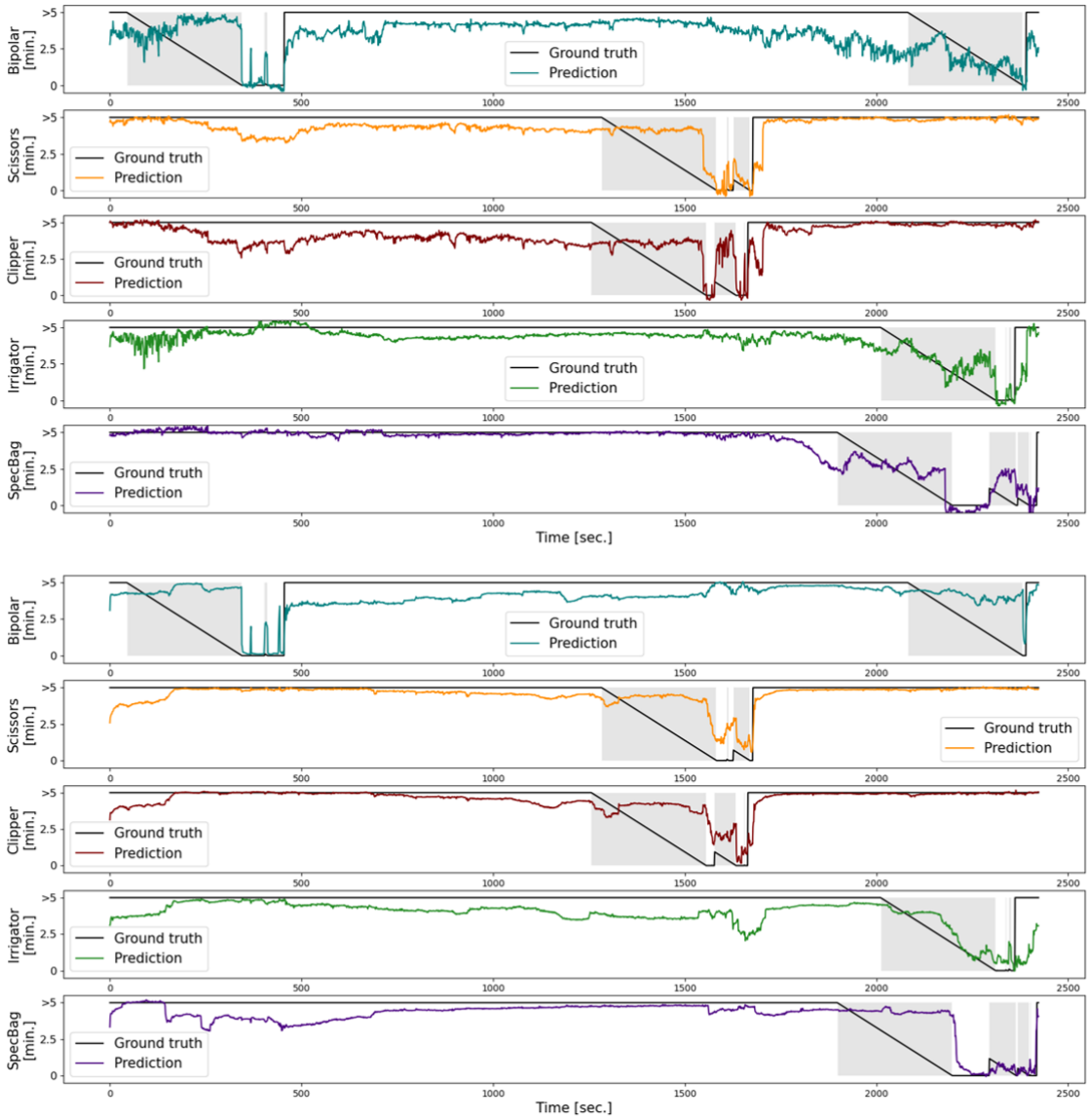


Figure 5.1: Instrument anticipation results. Upper part: result of our IIA-Net. Bottom part: result of the baseline [63]

Here, we utilize the plotting strategy mentioned before to show the sample results of the instrument anticipation and phase instrument. In both figures, the grey areas denote 'anticipating frames' ($0 < r(x_{T_{obs}}, \tau/\alpha) < h$), which are used to calculate inMAE. And our model is shown in the upper sub-figure, the baseline's result is shown in the bottom one. All sample results is based on the horizon as 5 min ($h = 5$).

As shown in the Figure. 5.1, our work performs better than the baseline. Specifically, it

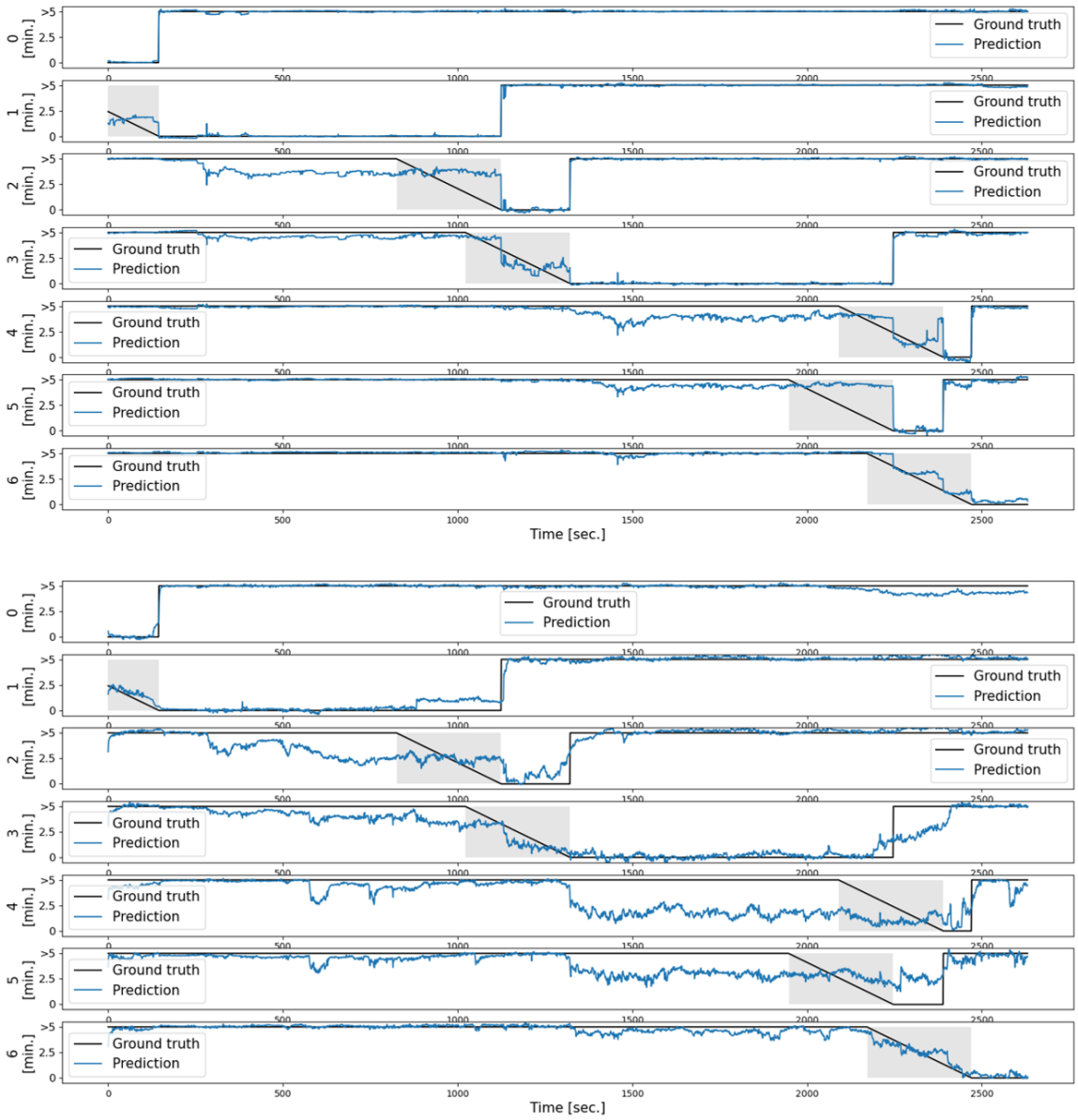


Figure 5.2: Phase anticipation result. Upper part: result of our IIA-Net. Bottom part: result of the baseline [63]

anticipates the occurrence bipolar and specimen bag correctly, while the [63] fails. In the baseline model, it even does not predict any trending of occurrence. But our model follows the decreasing trend in the 'anticipating' ares, which means that the system knows that the tool is about to happen. According to the scissor's anticipation, we can see that the prediction value drop sharply when it close to the occurrence. This is consistent to our previous analysis that our model is more sensitive to the frames that is highly close to the occurrence and achieves a better eMAE performance.

As shown in the Figure. 5.2, this work is comparable to the baseline but more stable. Compared to [63] our model (upper) achieves smoother and reliable prediction in terms of 'background frames'($r(x_{T_{obs}}, \tau/\alpha) = h$) and 'anticipating frames'($0 < r(x_{T_{obs}}, \tau/\alpha) < h$). It also give less false alert than [63] when the surgery is not ready to jump into the next phase. Also, it is notable that the baseline model tends to give the surgeon an occurrence signal when the next phase is far away. This could be ambiguous for the decision making in the online scenario. Also, the baseline shows the same occurrence pattern for the phase 4 and phase 4, which means that the model can not distinguish the triggering activity of different phases.

5.1.4 Earliest Consistency Criteria

To evaluate the consistency of the prediction and how early the system can properly anticipate the future, we propose the earliest consistency criteria to evaluate the model's performance. Since current prediction is mostly unstable and the frequency is large, only picking up the earliest proper prediction's timestamp is not appropriate to determine model's ability of early predicting. Therefore, we choose to visualize the predictions based on ea criteria, as shown in the Figure. 5.3. Here, we choose 0.5 min as the error threshold and select the worst result and the best result. A better anticipation result should contain more accepted predictions, i.e., error difference under the threshold and the predictions should be stable. Also, the system should be aware of the occurrence at the beginning of the ending of former instrument/phase. The figure plotted by the ea criteria contains these information, helping researchers to determine which is good result.

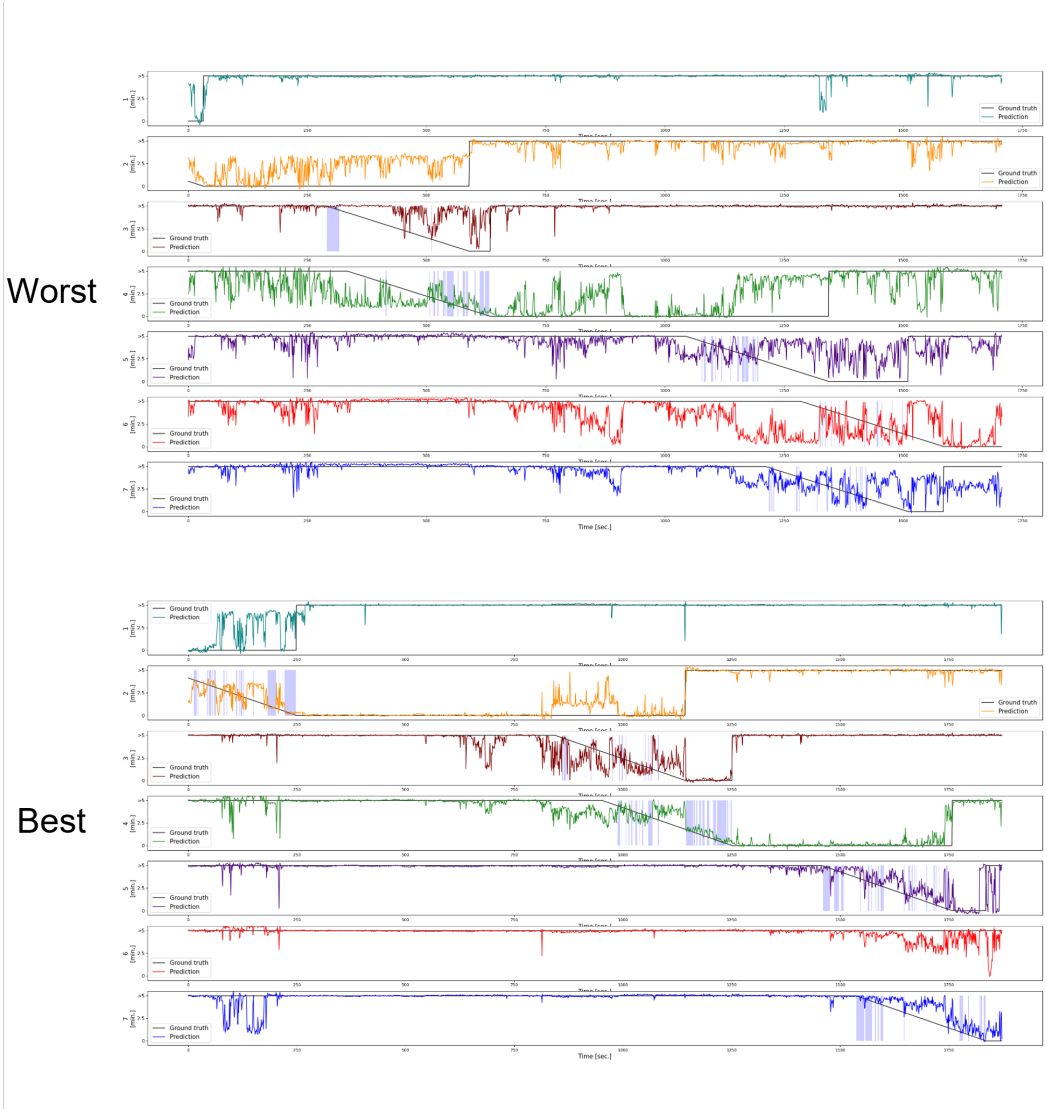


Figure 5.3: Phase anticipation result using the earliest consistency (ea) criteria. We select the worst anticipation result and compare with the best one to evaluate if the ea criteria can represent how well the system performs.

Table 5.9: Effect of IIM and MS-TCN on different feature extraction models for instrument anticipation. We report the inMAE averaging over instrument types in minutes per metric when $h = 5$ min. T: tool signal feature; P: phase signal feature; IIM: interaction feature from instrument interaction module.

	ResNet50	ResNet50+T+P	ResNet50+T+P+IIM	Baseline [63]
No MS-TCN	1.99	1.79	1.57	1.75
1 Stage	1.62	1.57	1.42	
2 Stages	1.59	1.45	**1.40**	
3 Stages	1.53	1.60	1.48	

Table 5.10: Effect of IIM and MS-TCN on different feature extraction models for instrument anticipation. We report the eMAE averaging over instrument types in minutes per metric when $h = 5$ min. T: tool signal feature; P: phase signal feature; IIM: interaction feature from instrument interaction module.

	ResNet50	ResNet50+T+P	ResNet50+T+P+IIM	Baseline [63]
No MS-TCN	4.06	3.58	2.51	2.68
1 Stage	3.74	3.29	2.22	
2 Stages	3.67	3.23	**2.14**	
3 Stages	3.64	3.31	2.15	

5.1.5 Horizons

The horizon uses truncation so that the model will not consider the remaining time predictions that are infinite far from the occurrence. The horizon should be the same when training and testing the model for a fair comparison. For example, if we train the model with the horizon as 5 and test it with the horizon as 2, it does not achieve the optimal result compared to the model trained and tested with $h = 2$. However, in real-world testing, the horizon is unseen by the user, meaning the horizon is dependent on the user's experience. Therefore, when performing the user study, the horizon setting up is also a factor that influences the system performance.

5.2 Effect of IIM and Stages in MS-TCN

We conduct ablative testing to compare different feature extraction models, ResNet50 [27], ResNet50 with instrument and phase features, ResNet50 with all added features, to identify a suitable feature extractor for our model. Additionally, we conduct experiments with different numbers of MS-TCN stages to determine which architecture is best able to capture temporal patterns.

As shown in Table. 5.9 and Table. 5.10, the ResNet50 with all features outperforms ResNet50 across the board with improvements ranging from 1.99 to 1.57 in inMAE and 4.06 to 2.51 in pMAE for surgical instrument anticipation. This increase can be attributed to the improved representation by our designed features. Among the features that we added, the IIM makes more contribution than the instrument and phase signals. This suggests that modeling the interactions signifies the surgeon's intention and the occurrence of the next situation. Interestingly, ResNet50 with all added feature achieves a comparable result with the baseline model [63] even without any temporal modeling.

Table. 5.9 and Table. 5.10 also highlight the substantial performance improvement achieved by the MS-TCN refinement stages. Those results demonstrate the ability of MS-TCN to improve the performance of any feature extractor. All feature extractors achieve higher performance when only 1 stage is used. However, 2 stages model outperforms 3 stages model. This could indicate that 3 stages of refinement lead to overfitting on the training set for the limited amount of data.

5.3 Semantic Segmentation

The input of IIM are two branches, segmentation map for the instrument-surrounding relation and tool detection for the the instrument-instrument relation. Both of the them are extracted from the model trained on the other public dataset without any finetuning upon the Cholec80. Therefore, it is unknown if the segmentation and detection provides a solid basis for the future feature extraction. To address this problem, we evaluate the performance of semantic segmentation extraction and instrument detection by visualizing the output predictions.

As shown in the Figure. 5.4, we visualize the semantic map from frames in video 20.

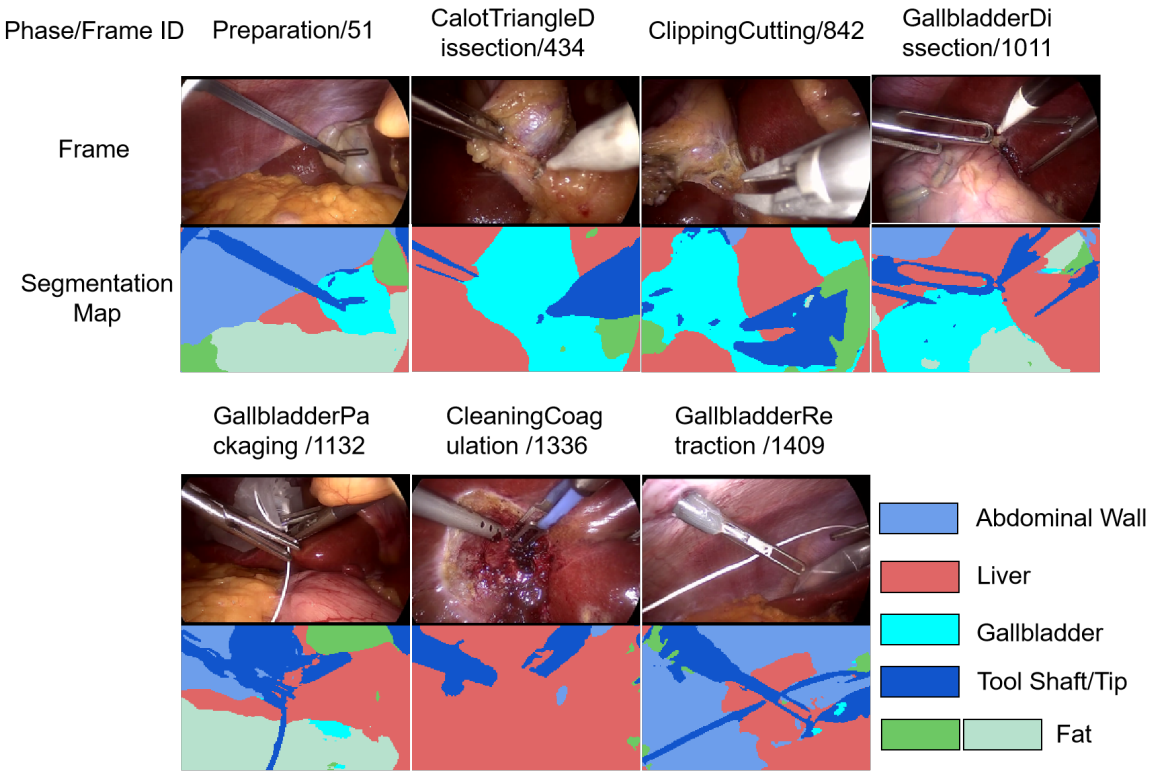


Figure 5.4: Semantic Segmentation Results on Cholec 80 dataset. We randomly select one frame from each phase to show the visualizatoin.

Specifically, we sample one frame for one phase and demonstrate how semantic map changes when phase changes. To evaluate its performance consistency, we select one frame from each phase. Here, we consider four categories of anatomical structures, i.e., abdominal wall, liver, gallbladder and fat. The visualization shows a promising result on distinguishing the gallbladder from the liver. The fat is the challenging structure to indicate in segmentation map because of its variance and random distribution in the body. The frame 1011 from Gallbladder Dissection shows that the abdominal wall is incorrectly classified as fat. For the surgical instrument segmentation, it shows a generally decent performance because the painting is consistent and all tools are correctly detected. Even there is still a lot improvement space, we show that the semantic map is effective to provide the instrument's surrounding information.

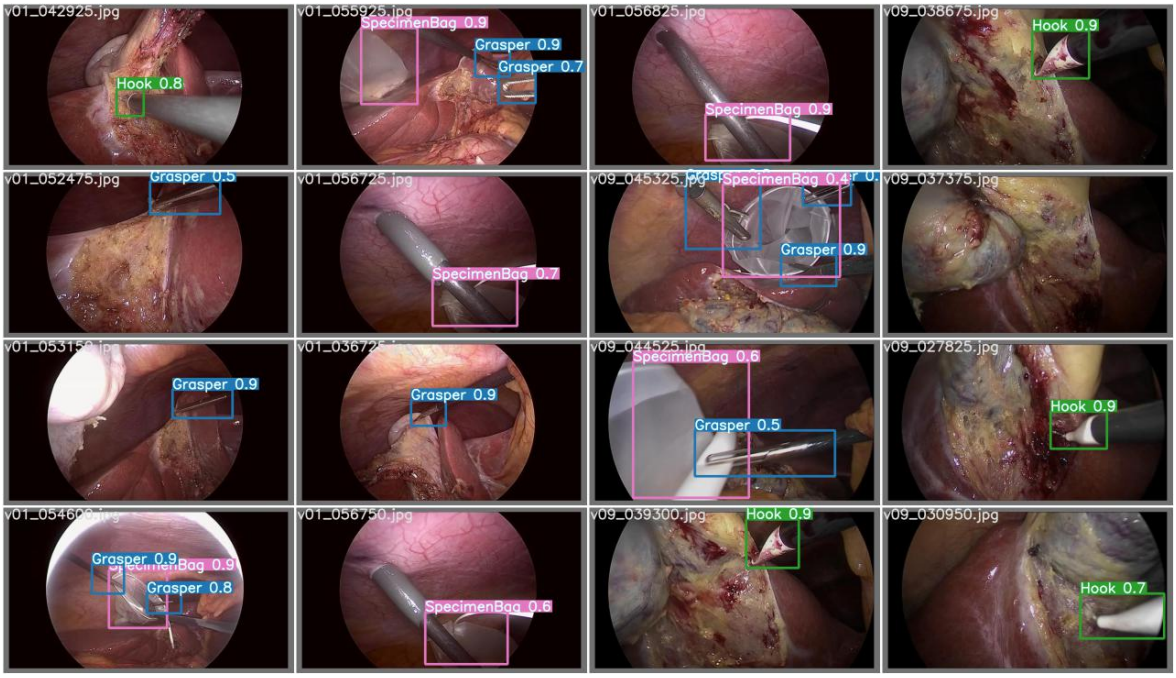


Figure 5.5: Tool detection results on Cholec 80 dataset.

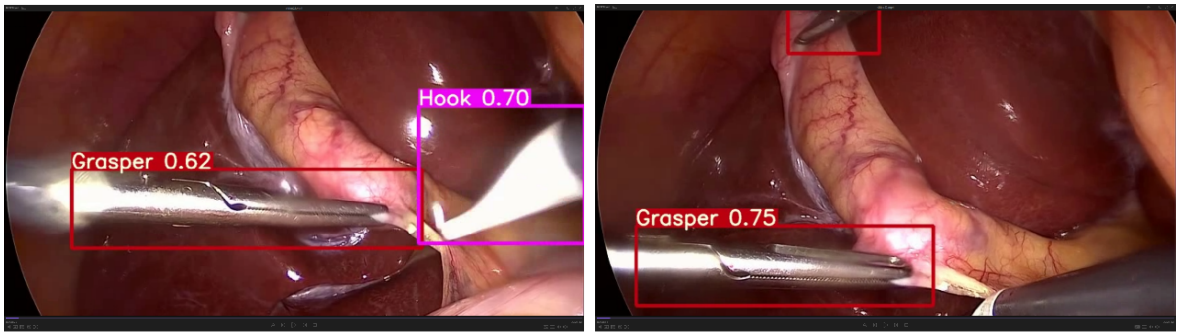


Figure 5.6: Inconsistent tool detection results. The time difference is 2 seconds but the detection of the hook is missed.

5.4 Tool Detection

To evaluate the generalization ability of detection from detection training set to the Cholec 80 dataset, we plot the bounding boxes upon the randomly selected images. As shown in the 5.5, the surgical instrument detection is nearly perfect. All the tools are detected correctly and the focus is only on the tool tip instead of the tool shift. However, the YOLO network is a 2D image detection approach and is performed frame by frame, leading to temporal inconsistency along time axis. The YOLO network is used in the 1 frame per second setting, where the model performs the inference once per second. For example, as shown in the 5.6, the instrument detection sometimes suddenly disappear in a short duration. This problem might decrease the feature's representational ability and degrade the performance.

5.5 Correlation

5.5.1 Instrument Instrument Correlation

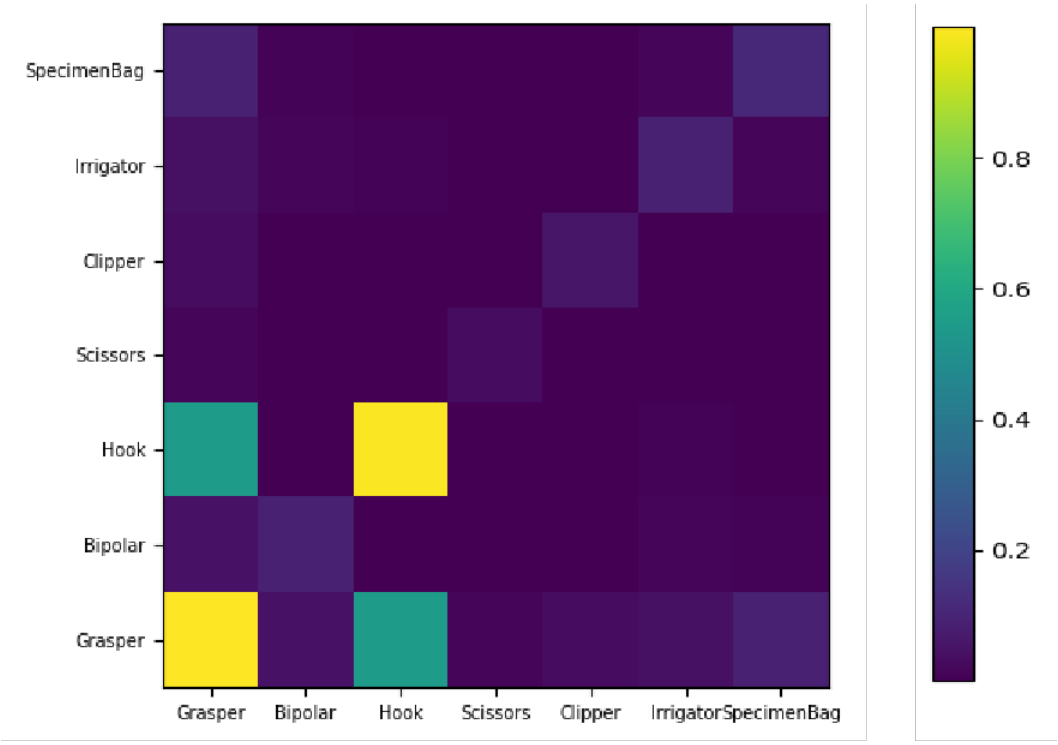


Figure 5.7: Correlation between instruments and instruments.

To support our hypothesis that the instruments' current presence is important for the surgical

anticipation, we calculate and visualize the correlation between instruments. As shown in the Figure. 5.7, the diagonal values shows the present frequency of each instruments, and the grasper is the most frequent instrument during the surgery. Also, we can see that the strong correlation exist between the grasper and hook. Also, the grasper almost always has the relation with the other instruments. This validate our idea about calculating instrument relation. Also, we show that the instrument presence is an effective signal to our anticipation task, and the introducing of recognition model is needed.

5.5.2 Instrument Phase Correlation

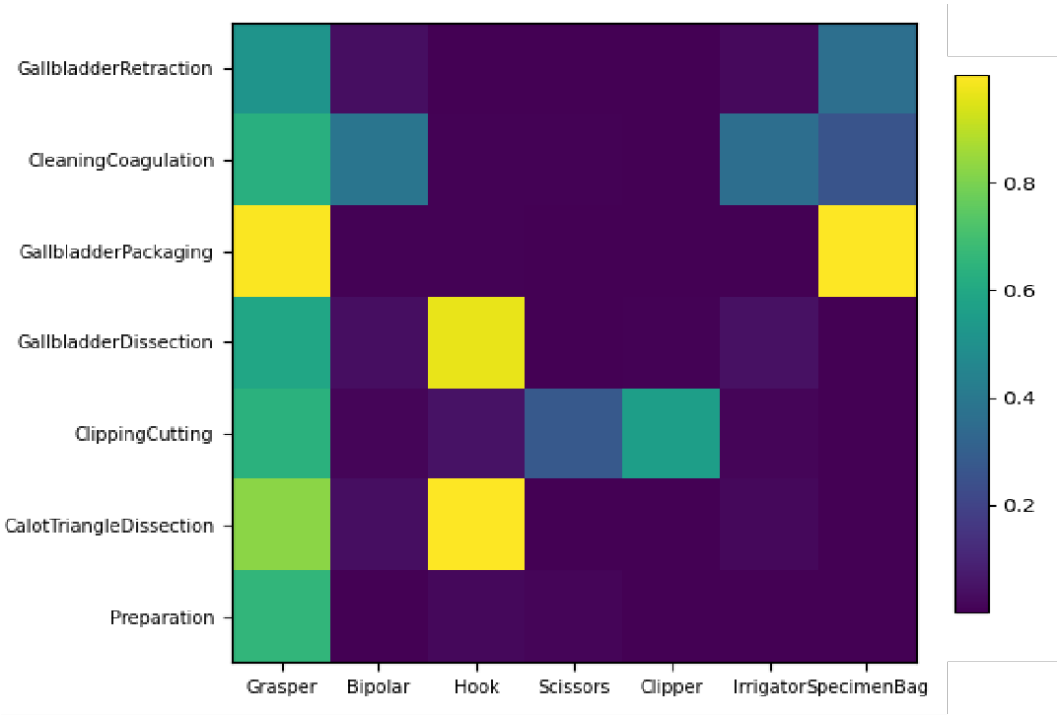


Figure 5.8: Correlation between instruments and phases

We also calculate the instrument-phase correlation to support our hypothesis that the phase' current presence is important for the surgical anticipation, we calculate and visualize the correlation between instruments and phases. As shown in the 5.8, the grasper has the the similar correlation value among all phases. This is because the grasper happens the most frequently, confirming our analysis in the previous section. There are other instrument-phase combinations showing the strong correlation. For example, the values of Hook-CalotTriangleDissection and Hook-GallbladderDissection show that hook is a mandatory

instrument and has to be used in these phases. For the specimen bag, even it shows strong correlation with the GallbladderPacking phase, it is also correlated with the CleaningCoagnulation and GallbladderRetraction phases. This is because the surgeon usually introduce the specimen bag at the end of the surgery and sometimes leave it aside to handle the complications, such as bleeding. Therefore, the specimen bag can last for a while until the surgeon really decide to pack the gallbladder and retract it. This correlation matrix indicates the motivation of recognition model's introduction and provide the insight for our multi-task learning design.

5.6 Phase Transition

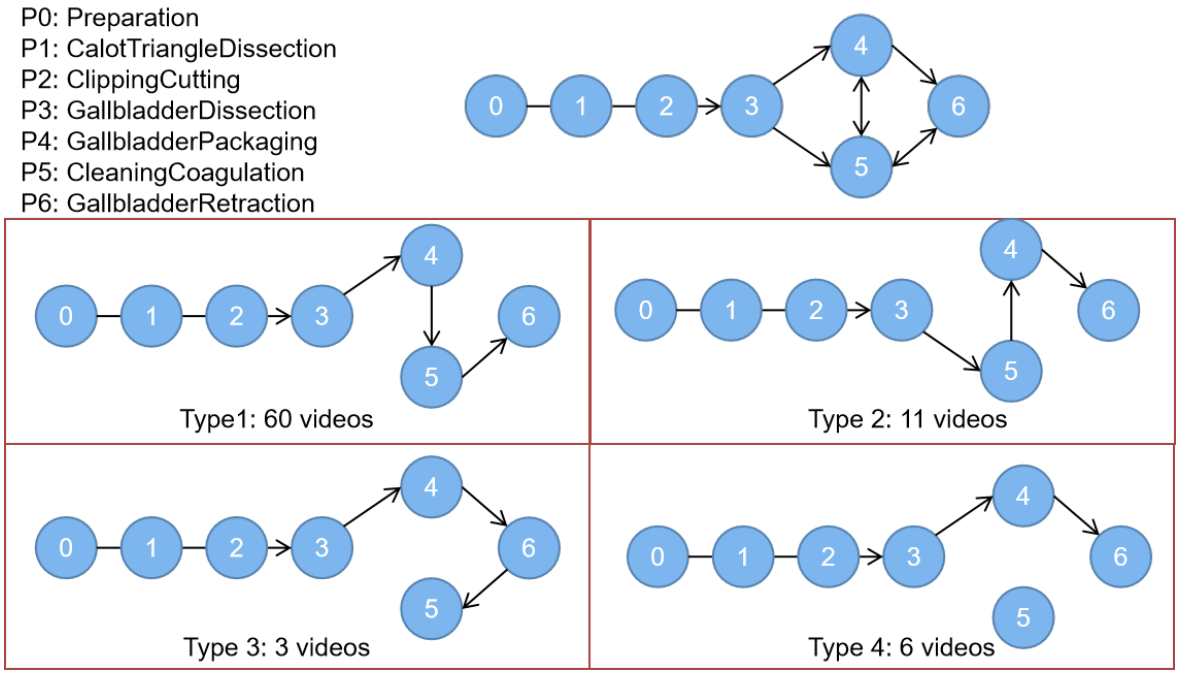


Figure 5.9: Phase transition. There are 4 categories of sequential patterns and the type 1 occupies the most in the Cholec 80 dataset.

In this section, we categorize the surgery types into four kinds by considering different phase transition types during the surgery. As shown in the Figure. 5.9, the general form of surgical workflow is predefined but with a flexibility among phase 3,4,5, and 6. This is because the coagulation cleaning and gallbladder retraction is highly depends on surgeon's intention and the patient's current status. In this section, we base on the Cholec 80 dataset and summarize

4 types of surgeries. We can see that the first type occupies the most of the surgeries and the others only have little amount samples. This could provide an insight for our future work about solving the long-tail distribution problem.

5.7 Troubleshoot the Model

If the instrument-interaction module is not working in the new environment, the ensembling strategy could be explored to reduce its effect. Specifically, we can train multiple models using different subsets of the features, similar to the random forest. The final prediction is the average voting from different models. This makes sure the model is not overfitting to a specific feature, such as the instrument-interaction features, thus leading to significant degradation when this feature does not work. Also, we can simply mask out the feature from IIM and retrain the temporal model. However, this will make the retrained model less powerful and might not be feasible for the application.

5.8 Limitations

The primary limitation of our current experimental setup is the incorporation of tool and phase signals. Specifically, we train the network using the signals from human annotation instead of recognition models. In real-world scenarios where this ground-truth is not available, the model's performance will likely be reduced. We conjecture, however, the degradation will be minimal using recent models which show superior performance for tool and phase recognition (95% and 88% of accuracy for tool and phase recognition). Also, [10] shows that the predicted phase signal is consistent and smooth not only within one phase, but also for the often ambiguous phase transitions. This means our IIA-Net will likely make a reliable prediction even with predicted tool and phase signals. In the newest experiment, we train the network using the signals from the recognition models, and the experimental results shown in this thesis is based on the predicted phase/tool presence signals instead of the ground truth.

Chapter 6

CONCLUSION

The surgical workflow anticipation is an unexplored task for the computer-assisted surgery system. Even if is a lot of works based on the deep neural networks (DNN), the task formulation is still not defined clearly. Nowadays, two main categories of solutions are proposed, segmentation sequence prediction and real time regression. In this work, we balance between these two formulations by considering the computational cost, running time performance, and field of view. And we opt for the real time regression form because it is more practical for the real-world surgical scenario.

This thesis presented a new instrument interaction aware deep neural network, named IIA-Net, to anticipate the surgical instrument/phase occurrence 2/3/5 minutes ahead. IIA-Net is an efficient neural network that can be applied to the online surgery scenario to support many relevant tasks. In this work, we target two main components, spatial feature extractor and temporal modeling method to build our model. Firstly, in order to accurately understand the intention of the surgeons, such as the initialization of gallbladder retraction, we propose a novel feature extraction module instrument-interaction module (IIM) to capture the instrument-instrument and instrument-surrounding interaction. This feature extraction module is proved to be effective and low cost since it needs the pre-extracted segmentation map and instrument detection. Secondly, we propose to utilize the MS-TCN as the temporal modeling method to fine-tune the result. To allow our IIA-Net applied into the online surgery scenario, we alter the convention 1D convolution into causal convolution operation by padding and shifting the temporal sequence. Also, we propose to utilize the two-stage training strategy to separately train the feature extractor and temporal model with the limited computational resources. Experimental analysis exhibit that our proposed instrument-interaction aware model achieves a higher standard than any other state-of-the-art methods with semantic and detection prior information.

6.1 Accomplishment

We propose an efficient deep learning model for temporal surgical workflow analysis, namely the IIA-Net, which incorporates existing surgical workflow analysis methods, i.e., tool detection, phase recognition, laparoscopic image segmentation, and outperforms previous works. It shows that the interaction relationship during spatial feature extraction is effective to resolve surgical workflow anticipation. Without temporal training, our model is a strong baseline for the following 2D works. Furthermore, temporal modelling using a MS-TCN with causal and dilated convolution handles full temporal resolution of time series, fitting extreme long laparoscopic workflow well. Its large receptive field captures distant as well recent observations. Our multi-task learning schema provides a potential direction to jointly perform instrument and phase anticipation.

6.2 Real World Application

This work's motivation is to benefit the current computer-assisted surgery system, because this proposed architecture has can generate a useful peripheral signal, indicating the remaining time until the next occurrence of certain instrument/phase. This signal offers two-fold prominent benefits for clinicians. Firstly, our reliable tool anticipation provides a useful reference to robotic systems in decision making. It can support the surgeon to decide when to go into the next phase and when to introduce next instrument for the current situation. Secondly, accurately anticipating tools such as the irrigator can help the early detection and even prevention of potential complications, for example, massive haemorrhage. Thirdly, it allows real-time instruction for automated surgical coaching therefore increasing patient safety and reducing surgical errors. The anticipation of surgical phases can also provide vital input for optimizing communication in the operating room (OR).

6.3 Future Work

Our method can estimate the occurrence of next instrument/phase occurrence; however, there is still an improvement space for the future works. Here I will discuss the potential directions from the dataset and computing cost perspectives. Firstly, on-device training as the solution

to deploy the training process to the edge device, can save the energy and computing resource demanding. Also, it can help to develop a personalized neural network by combining the edge device and surgery's recording device, eliminating the privacy and ethical issue in medical domain. Secondly, patient-wise fine-tuning is needed. Even the neural network has achieved a promising result, it is occurred with the generalization issue when the dataset is enlarged and new data samples are introduce from different patients. Therefore, the network's fine-tuning for each patient is needed. Also, given the fact that manually annotation is expensive, a self-supevised training strategy should be included for a better modeling ability.

Another improvement lies in improving the representational ability of the spatial feature extraction process. The multi-stage temporal convolutional network is a light-weight model and can handle complicated sequence modeling efficiently. Therefore, it is needed to change from the feature perspective. For example, combining surgical expert's prior experience to design the feature extraction blocks, similar to the instrument-interaction module in this work, is a potential direction.

BIBLIOGRAPHY

[1] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.

[2] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5343–5352, 2018.

[3] Blake C Alkire, Nakul P Raykar, Mark G Shrime, Thomas G Weiser, Stephen W Bickler, John A Rose, Cameron T Nutt, Sarah LM Greenberg, Meera Kotagal, Johanna N Riesel, et al. Global access to surgical care: a modelling study. *The Lancet Global Health*, 3(6):e316–e323, 2015.

[4] Kathuria Ayoosh. What's new in yolo v3? `https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b` Accessed April 23, 2018.

[5] Yutong Ban, Guy Rosman, Thomas Ward, Daniel Hashimoto, Taisei Kondo, Hidekazu Iwaki, Ozanan Meireles, and Daniela Rus. Surgical prediction gan for events anticipation. *arXiv preprint arXiv:2105.04642*, 2021.

[6] Loubna Bouarfa, Pieter P Jonker, and Jenny Dankelman. Discovery of high-level tasks in the operating room. *Journal of biomedical informatics*, 44(3):455–462, 2011.

[7] Zang-Hee Cho, Joie P Jones, and Manbir Singh. *Foundations of medical imaging*. Wiley New York, 1993.

[8] Kevin Cleary, Ho Young Chung, and Seong K Mun. Or2020 workshop overview: operating room of the future. In *International Congress Series*, volume 1268, pages 847–852. Elsevier, 2004.

[9] Kevin Cleary and Terry M Peters. Image-guided interventions: technology review and clinical applications. *Annual review of biomedical engineering*, 12:119–142, 2010.

[10] Tobias Czempiel, Magdalini Paschali, Matthias Keicher, Walter Simson, Hubertus Feussner, Seong Tae Kim, and Nassir Navab. Tecno: Surgical phase recognition with multi-stage temporal convolutional networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 343–352. Springer, 2020.

[11] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564, 2016.

[12] Samuel Dupond. A thorough review on the current advance of neural network structures. *Annual Reviews in Control*, 14:200–230, 2019.

[13] Marzieh Ershad, Zachary Koesters, Robert Rege, and Ann Majewicz. Meaningful assessment of surgical expertise: Semantic labeling with data and crowds. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 508–515. Springer, 2016.

[14] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019.

[15] Ali Farhadi and Joseph Redmon. Yolov3: An incremental improvement. *Computer Vision and Pattern Recognition, cite as*, 2018.

[16] AGMLS Fernandez, R Bertolami H Bunke, and J Schmiduber. A novel connectionist system for improved unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5), 2009.

[17] Germain Forestier, Florent Lalys, Laurent Riffaud, D Louis Collins, Jurgen Meixensberger, Shafik N Wassef, Thomas Neumuth, Benoit Goulet, and Pierre Jannin. Multi-site study of surgical practice in neurosurgery based on surgical process models. *Journal of Biomedical Informatics*, 46(5):822–829, 2013.

[18] Germain Forestier, François Petitjean, Laurent Riffaud, and Pierre Jannin. Automatic matching of surgeries to predict surgeons' next actions. *Artificial intelligence in medicine*, 81:3–11, 2017.

[19] Stefan Franke and Thomas Neumuth. Adaptive surgical process models for prediction of surgical work steps from surgical low-level activities. In *6th Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI) at the 18th International Conference on Medical Image Computing and Computer Assisted Interventions (MICCAI), Munich, Germany*, 2015.

[20] Isabel Funke, Sören Torge Mees, Jürgen Weitz, and Stefanie Speidel. Video-based surgical skill assessment using 3d convolutional neural networks. *International journal of computer assisted radiology and surgery*, 14(7):1217–1225, 2019.

[21] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. *arXiv preprint arXiv:1707.04818*, 2017.

[22] Yixin Gao, S Swaroop Vedula, Carol E Reiley, Narges Ahmidi, Balakrishnan Varadarajan, Henry C Lin, Lingling Tao, Luca Zappella, Benjamın Béjar, David D Yuh, et al. Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In *MICCAI workshop: M2cai*, volume 3, page 3, 2014.

[23] Atul Gawande. Two hundred years of surgery. *New England Journal of Medicine*, 366(18):1716–1723, 2012.

[24] Atul A Gawande, Eric J Thomas, Michael J Zinner, and Troyen A Brennan. The incidence and nature of surgical adverse events in colorado and utah in 1992. *Surgery*, 126(1):66–75, 1999.

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[26] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Mark A Healey, Steven R Shackford, Turner M Osler, Frederick B Rogers, and Elizabeth Burns. Complications in surgical patients. *Archives of surgery*, 137(5):611–618, 2002.

[29] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[30] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[31] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[32] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[33] Amy Jin, Serena Yeung, Jeffrey Jopling, Jonathan Krause, Dan Azagury, Arnold Milstein, and Li Fei-Fei. Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 691–699. IEEE, 2018.

[34] Yueming Jin, Qi Dou, Hao Chen, Lequan Yu, Jing Qin, Chi-Wing Fu, and Pheng-Ann Heng. Sv-rcnet: workflow recognition from surgical videos using recurrent convolutional network. *IEEE transactions on medical imaging*, 37(5):1114–1126, 2017.

[35] B-H Juang. On the hidden markov model and dynamic time warping for speech recognition—a unified view. *AT&T Bell Laboratories Technical Journal*, 63(7):1213–1243, 1984.

[36] Timothy N Judkins, Dmitry Oleynikov, and Nick Stergiou. Objective evaluation of expert and novice performance during robotic surgical training tasks. *Surgical endoscopy*, 23(3):590–597, 2009.

[37] AK Kable, RW Gibberd, and AD Spigelman. Adverse events in surgical patients in australia. *International Journal for Quality in Health Care*, 14(4):269–276, 2002.

[38] Darko Katić, Anna-Laura Wekerle, Fabian Gärtner, Hannes Kenngott, Beat Peter Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Knowledge-driven formalization of laparoscopic surgeries for rule-based intraoperative context-aware assistance. In *International Conference on Information Processing in Computer-Assisted Interventions*, pages 158–167. Springer, 2014.

[39] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-conditioned action anticipation in one shot. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9925–9934, 2019.

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[41] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[42] Ulrich Klank, Nicolas Padoy, Hubertus Feussner, and Nassir Navab. Automatic feature generation in endoscopic images. *International Journal of Computer Assisted Radiology and Surgery*, 3(3):331–339, 2008.

[43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[44] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

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

[46] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[47] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2019.

[48] Ke Liang, Yuan Xing, Jianmin Li, Shuxin Wang, Aimin Li, and Jinhua Li. Motion control skill assessment based on kinematic analysis of robotic end-effector movements. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 14(1):e1845, 2018.

[49] Suhas Lohit, Qiao Wang, and Pavan Turaga. Temporal transformer networks: Joint learning of invariant and discriminative time warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12426–12435, 2019.

[50] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

[51] Tahmida Mahmud, Mahmudul Hasan, and Amit K Roy-Chowdhury. Joint prediction of activity labels and starting times in untrimmed videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5773–5782, 2017.

[52] Lena Maier-Hein, Swaroop S Vedula, Stefanie Speidel, Nassir Navab, Ron Kikinis, Adrian Park, Matthias Eisenmann, Hubertus Feussner, Germain Forestier, Stamatia

Giannarou, et al. Surgical data science for next-generation interventions. *Nature Biomedical Engineering*, 1(9):691–696, 2017.

[53] JA Martin, Glenn Regehr, Richard Reznick, Helen Macrae, John Murnaghan, Carol Hutchison, and M Brown. Objective structured assessment of technical skill (osats) for surgical residents. *Journal of British Surgery*, 84(2):273–278, 1997.

[54] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Fast scene understanding for autonomous driving. *arXiv preprint arXiv:1708.02550*, 2017.

[55] Chinedu Innocent Nwoye, Cristians Gonzalez, Tong Yu, Pietro Mascagni, Didier Mutter, Jacques Marescaux, and Nicolas Padoy. Recognition of instrument-tissue interactions in endoscopic videos via action triplets. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 364–374. Springer, 2020.

[56] Nicolas Padoy. Machine and deep learning for workflow recognition during surgery. *Minimally Invasive Therapy & Allied Technologies*, 28(2):82–90, 2019.

[57] Nicolas Padoy, Tobias Blum, Seyed-Ahmad Ahmadi, Hubertus Feussner, Marie-Odile Berger, and Nassir Navab. Statistical modeling and recognition of surgical workflow. *Medical image analysis*, 16(3):632–641, 2012.

[58] Micha Pfeiffer, Isabel Funke, Maria R Robu, Sebastian Bodenstedt, Leon Strenger, Sandy Engelhardt, Tobias Roß, Matthew J Clarkson, Kurinchi Gurusamy, Brian R Davidson, et al. Generating large labeled data sets for laparoscopic image processing tasks using unpaired image-to-image translation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 119–127. Springer, 2019.

[59] Behnaz Poursartip, Marie-Eve LeBel, Rajni V Patel, Michael D Naish, and Ana Luisa Trejos. Analysis of energy-based metrics for laparoscopic skills assessment. *IEEE Transactions on Biomedical Engineering*, 65(7):1532–1542, 2017.

[60] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[61] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[62] Zarreen Naowal Reza. *Real-time automated weld quality analysis from ultrasonic B-scan using deep learning*. PhD thesis, University of Windsor (Canada), 2019.

[63] Dominik Rivoir, Sebastian Bodenstedt, Isabel Funke, Felix von Bechtolsheim, Marius Distler, Jürgen Weitz, and Stefanie Speidel. Rethinking anticipation tasks: Uncertainty-aware anticipation of sparse surgical instrument usage for context-aware assistance. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 752–762. Springer, 2020.

[64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[65] Mohammad Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Basura Fernando, Lars Petersson, and Lars Andersson. Encouraging lstms to anticipate actions very early. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 280–289, 2017.

[66] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014.

[67] Laura Sevilla-Lara, Yiyi Liao, Fatma Güney, Varun Jampani, Andreas Geiger, and Michael J Black. On the integration of optical flow and action recognition. In *German Conference on Pattern Recognition*, pages 281–297. Springer, 2018.

[68] Yarden Sharon, Thomas Sean Lendvay, and Ilana Nisky. Instrument orientation-based metrics for surgical skill evaluation in robot-assisted and open needle driving. *arXiv preprint arXiv:1709.09452*, 2017.

[69] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014.

[70] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[71] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020. IEEE, 2018.

[72] Ana Luisa Trejos, Rajni V Patel, Richard A Malthaner, and Christopher M Schlachta. Development of force-based metrics for skills assessment in minimally invasive surgery. *Surgical endoscopy*, 28(7):2106–2119, 2014.

[73] Andru P Twinanda, Didier Mutter, Jacques Marescaux, Michel de Mathelin, and Nicolas Padoy. Single-and multi-task architectures for surgical workflow challenge at m2cai 2016. *arXiv preprint arXiv:1610.08844*, 2016.

[74] Andru P Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel De Mathelin, and Nicolas Padoy. Endonet: a deep architecture for recognition tasks on laparoscopic videos. *IEEE transactions on medical imaging*, 36(1):86–97, 2016.

[75] Andru Putra Twinanda, Gaurav Yengera, Didier Mutter, Jacques Marescaux, and Nicolas Padoy. Rsdnet: Learning to predict remaining surgery duration from laparoscopic videos without manual annotations. *IEEE transactions on medical imaging*, 38(4):1069–1078, 2018.

[76] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[77] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *international conference on machine learning*, pages 3560–3569. PMLR, 2017.

[78] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 98–106, 2016.

[79] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 606–615, 2018.

[80] Ziheng Wang and Ann Majewicz Fey. Deep learning with convolutional neural network for objective skill evaluation in robot-assisted surgery. *International journal of computer assisted radiology and surgery*, 13(12):1959–1970, 2018.

[81] Thomas G Weiser, Alex B Haynes, George Molina, Stuart R Lipsitz, Micaela M Esquivel, Tarsicio Uribe-Leitz, Rui Fu, Tej Azad, Tiffany E Chao, William R Berry, et al. Estimate of the global volume of surgery in 2012: an assessment supporting improved health outcomes. *The Lancet*, 385:S11, 2015.

[82] Thomas G Weiser, Scott E Regenbogen, Katherine D Thompson, Alex B Haynes, Stuart R Lipsitz, William R Berry, and Atul A Gawande. An estimation of the global volume of surgery: a modelling strategy based on available data. *The Lancet*, 372(9633):139–144, 2008.

[83] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[84] Aneeq Zia and Irfan Essa. Automated surgical skill assessment in rmis training. *International journal of computer assisted radiology and surgery*, 13(5):731–739, 2018.