AUTOMATED BENCHMARKING OF SURGICAL SKILLS USING MACHINE LEARNING

A Dissertation
Presented to
The Academic Faculty

By

Aneeq Zia

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

December 2018

AUTOMATED BENCHMARKING OF SURGICAL SKILLS USING MACHINE LEARNING

Approved by:

Dr. Irfan Essa, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Patricio Vela
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Thomas Ploetz
School of Interactive Computing
Georgia Institute of Technology

Dr. Anthony Jarc
Research and Data Science
Intuitive Surgical

Dr. David Anderson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: October 31, 2018

To my family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this PhD research is to design and develop automated systems for evaluation of surgical skills in order to reduce manual assessment by experts and help surgical trainees to move through their learning curves much faster.

Surgical trainees are required to acquire specific skills during the course of their residency before performing real surgeries. Surgical training involves constant practice of skills and seeking feedback from supervising surgeons, who generally have a packed schedule. The process of manual assessment makes the whole training cycle extremely cumbersome and inefficient. Having automated assessment systems for surgical training can be of great value to medical schools and teaching hospitals.

A typical surgical trainee goes through multiple stages during their training programs. Most of them start with practicing basic skills of suturing and knot tying on foam boards/synthetic tissue. Then they go on to practicing on VR based consoles where they learn more advanced and clinical relevant skills but without real tissue involved. Once they have acquired the desired level of competency in the previous stages, they then practice on cadavers or pigs before moving on to performing surgeries on real patients. There is a need for automated assessment at every stage of surgical training.

This PhD research aims at developing machine learning based methods for assessment of surgical skills from basic tasks to complex robot-assisted procedures. Specifically, this thesis will aim to (1) develop novel motion based features for basic surgical skills assessment in open and robotic surgical training, (2) develop unsupervised and supervised methods for recognizing individual steps of complex robot-assisted (RA) surgical procedures, (3) generate automated score reports for RA surgical procedures, and (4) produce video highlights to indicate which parts of the surgical task most effected the final surgical skill score.

CHAPTER 1
INTRODUCTION

Surgical skill development, i.e., the process of gaining expertise in procedures and techniques required for professional surgery, represents an essential part of medical training. Acquiring high quality surgical skills is a time-consuming process that demands expert supervision and evaluation throughout all stages of the training procedure. However, the manual assessment of surgical skills poses a significant resource problem to medical schools and teaching hospitals and results in complications in executing and scheduling their day-to-day activities. In addition to the extensive time requirements, manual assessments are often subjective and domain experts do not always agree on the assessment scores

Surgery is a complex task and even basic surgical skills such as suturing and knot tying (that involve hand movements in a repetitive manner) require every surgical resident to go through training in order to master these basic skills before moving on to more complicated procedures. Considering the volume of trainees that need to go through basic surgical skills training along with the time consuming and subjective nature of manual evaluation, automated assessment of these basic surgical skills can be of tremendous benefit to medical schools and teaching hospitals.

Medical literature recognizes the need for objective surgical skill assessment in surgical training [1]. Yu et al. [2] have suggested evaluations from residents and interns who frequently supervise the students instead of the consultant surgeons who do not have the opportunity to directly observe the medical students. However, the subjectivity and time-consuming nature of these evaluations still cannot be ruled out.

For basic surgical tasks like suturing and knot tying in a training setup, structured grading systems such as the Objective Structured Assessment of Technical Skills (OSATS) [3] have been developed to reduce the subjectivity. Table 1.1 summarizes the OSATS scoring

Table 1.1: Summary of the OSATS scoring system [3]. The score is a Likert scale from levels 1-5 but the guidelines are provided only for levels 1, 3, and 5. The diversity of the criteria, lack of guidelines for all levels, and the need to manually observe each surgeon, makes the manual OSATS scoring a time consuming and challenging process.

Score	**Respect for tissue (RT)**	**Time and motion (TM)**	**Instrument handling (IH)**	**Suture handling (SH)**	**Flow of operation (FO)**	**Knowledge of procedure (KP)**	**Overall performance** (OP)
1	Unnecessary force on tissue, caused damage	Unnecessary moves	Inappropriate instrument use	Repeated entanglement, poor knot tying	Seemed unsure of next move	Insufficient knowledge	Very poor
2	–	–	–	–	–	–	–
3	Occasionally caused damage	Some unnecessary moves	Occasionally stiff or awkward	Majority of knots placed correctly	Some forward planning	Knew all important steps	Competent
4	–	–	–	–	–	–	–
5	Minimal tissue damage	Economy of movement	Fluid movements	Excellent suture control	Planned operation	Familiarity with all steps	Clearly superior

system. OSATS consists of seven generic components of operative skill that are marked on a 5 point Likert scale. OSATS criteria are diverse and depend on different aspects of motion. For instance, qualitative criteria such as "respect for tissue" depend on overall motion quality while sequential criteria such as "time and motion" and "knowledge of procedure" depend on motion execution order.

For more complex surgical training, like on the da Vinci robotic system, most of the objective assessment is based on efficiency metrics like economy of motion, speed, camera movement etc. However, unlike basic training where the surgeon is only performing a single task (e.g only suturing), clinical procedures being performed on robotic systems usually involve multiple steps and can take a few hours to complete. This makes the assessment of surgical skills even harder than that compared to basic training. Currently, intraoperative assessment has been limited to feedback from attendings and/or proctors. Aside from the qualitative feedback from experienced surgeons, quantitative feedback has remained abstract to the level of an entire procedure, such as total duration. Performance feedback for one particular task within a procedure can potentially be more helpful to direct opportunities of improvement. Similarly, statistics from the entire surgery may not be ideal to show an impact on outcomes. For example, one might want to closely examine the performance

of a single task if certain adverse outcomes are related to only that specific step of the entire procedure. Scalable methods to recognize automatically when particular tasks occur within a procedure are needed to generate these metrics to then provide feedback to surgeons or correlate to outcomes.

The growing need for automated assessment of surgical skills in various stages of training and practice motivated us to develop machine learning based methods that can help provide objective automated score-based feedback to surgeons. This work focuses on assessment of basic surgical skills like that of suturing and knot tying, and assessment in robot-assisted clinical procedures while tackling the problem of procedure segmentation in order to achieve that.

1.1 Challenges

Replicating the assessments provided by experts is not an easy task to achieve. There are many challenges that make this problem quite hard to solve using machine learning. A few of them are listed below.

1. **Disagreement between experts:** The subjective nature of assessment results in differences between scores that different experts give to the same trainee. This is mainly due to the fact that surgeons can vary significantly on their style of surgery and can perform the same task with competency in a very different way. Naturally, as a result, the trainee more near to their own style would be given better scores.

2. **Availability of data:** Any machine learning problem requires good amount of data to start with. Unfortunately, there are very few data sets available in the surgical domain, and those present are very small in size. Therefore, for our work, new data needs to be collected and annotated before machine learning based models can be developed.

3. **Huge variation in clinical procedures:** For assessment in clinical robot-assisted

procedures, the first problem to solve is that of segmenting the procedure into individual steps. This is an extremely challenging task by itself since each surgeon will have different kinds of motion and each person's anatomy will look different. There is nothing standard in such a task which requires very robust models to be developed that can take in all the information coming from the robotic system to recognize individual tasks. Since procedure segmentation is hard to fully solve, assessment on top of that becomes challenging as well since that is all dependent on how well the procedure segmentation works.

1.2 Organization of the thesis

This thesis is organized as follows: Chapter 2 provides a detailed review on previous literature revolving around surgical skill assessment and surgical activity recognition that provided the motivation behind this PhD work. In chapter 3, we introduce novel motion based features for surgical skill assessment using video and accelerometer data for open surgical training. Chapter 4 extends the evaluations of features proposed in Chapter 3 to robot-assisted (RA) basic surgical training tasks. The thesis then drives into the domain of more complex RA procedures and covers unsupervised and supervised surgical activity work proposed in Chapter 5 and 6, respectively. Chapter 7 uses the work of supervised activity recognition from Chapter 6 to propose an automated system for surgical performance report generation for RA procedures. Chapter 8 finishes the technical part of this thesis and covers methods proposed for video highlight generation in surgical procedures. A brief summary and possible future work are given at the end of this thesis.

CHAPTER 2
BACKGROUND

In this section we will look at some of the key works done in the field that motivated the work presented in this thesis. The first section of this chapters provides a survey on various papers presented in literature for surgical skills assessment. The next section gives a background on work done in surgical activity recognition, which is followed by conclusion. Table 2.1 summarizes some of the key works from both sections.

2.1 Surgical Skill Assessment

The problem of automated surgical skills assessment has recently seen some good progress [4, 5, 6, 7, 8, 9]. Pioneering efforts were based on robotic minimally invasive surgery (RMIS) and focused on gesture recognition and skill assessment using Hidden Markov Models [10, 11, 12]. These initial endeavors attempted to identify gestures or motion sequences for a specific surgical task. These gesture based methods were mostly used for surgical activity recognition and in some cases for surgical skill assessment.

For assessment of surgical skills in RMIS, one of the earlier works proposed a variant of HMM - sparse HMM [13]. Other works like [7] studied the differences in needle-driving movements and reported significant differences between beginner and expert surgeons. In [8], the authors proposed descriptive curve coding-common string model (DCC-CSM) for simultaneous surgical gesture recognition and skill assessment. Support Vector Machine (SVM) have also been used on basic metrics like time for completion, path length, speed etc, for skill evaluation [9]. [7] studied robotic surgical movements and reported significant difference in the needle-driving movements of experienced surgeons and novices. GEARS (Global Evaluative Assessment of Robotic Skills) is an assessment tool specifically developed to assess levels of robotic surgical expertise and is known to be consistent and reliable

as reported in [14]. More recently, some works have explored the use of crowd sourcing techniques to evaluate surgeon skill [15].

Despite advances in basic robotic surgical training, assessment of conventional surgical skills is done using OSATS [3] in medical schools and teaching hospitals (see table 1.1 for details on OSATS grading scheme). Some works based on automated assessment of the OSATS criteria for general surgical training have also been proposed recently. In [6], the authors introduced Augmented BoW (ABoW), in which time and motion are modeled as short sequences of events and the underlying local and global structural information is automatically discovered and encoded into BoW models. They classified surgeons into different skill levels based on the holistic analysis of time series data. In [4], the authors proposed Motion Texture (MT) analysis technique in which each video is represented as a multi-dimensional sequence of motion class counts to obtain a frame kernel matrix. The textural features derived from the frame kernel matrix are used for prediction of OSATS criteria. Although MT technique provided good OSATS prediction, it is computationally intensive ($N \times N$ sized frame kernel matrix for a video with N frames) and does not account for the sequential motion aspects in surgical tasks. A variant of MT, called Sequential Motion Texture (SMT) [5], encoded both the qualitative and sequential motion aspects.

The techniques mentioned above do provide encouraging results for video based OSATS-like surgical skill assessment. However, these studies use very few participants which limits their ability to capture the wide variation in surgical skills. An expert surgeon's hand motion might be more clean, distinct, ordered and sequential as compared to a non-expert and having more samples helps capture skills of varying levels. However, most of the works mentioned above have not tried to utilize the disorder and repetitiveness in motion for skill assessment. Also, they do not include studies on wearable motion sensing devices such as accelerometers that may provide precise motion information for surgical skills assessment.

Table 2.1: Key works on surgical skill assessment and surgical phase recognition. CNN: Convolutional Neural Network, SMT: Sequential Motion Texture, CRF: Conditional Random Field, BoW: Bag-of-Words, ABoW: Augmented Bag-of-Words, LDS: Linear Dynamical Systems, DTW: Dynamic Time Warping, CCA: Canonical Correlation Analysis, HMM: Hidden Markov Model

Reference	Technique	Phase	Analysis goal	Data
Dipietro (2016) [16]	RNN	Yes	Surgical gesture recognition	RMIS (only kinematic data from robotic surgery), 23 subjects
Twinanda (2016) [17]	CNN	Yes	Surgical tool detection and phase recognition	Laparoscopic cholecystectomy (endoscopic video), 13 subjects
Lea (2015) [18]	CRF	Yes	Surgical action segmentation and recognition	RMIS (both kinematic and video data from robotic surgery), 8 subjects
Sharma (2014) [4, 5]	MT,SMT	No	OSATS prediction, classification	General suturing task (only video data), 16 subjects
Tao (2013) [12]	CRF	Yes	Surgical gesture segmentation and recognition	RMIS (both kinematic and video data from robotic surgery), 8 subjects
Bettadapura (2013) [6]	ABoW	No	OSATS classification	General suturing task (only video data), 16 subjects
Haro (2012), Zapella (2013) [19, 20]	BoW, LDS	Yes	Surgical gesture recognition	RMIS (both kinematic and video data from robotic surgery), 8 subjects
Padoy (2012) [21]	DTW, HMM	Yes	Surgical phase recognition	Laparoscopic cholecystectomy (endoscopic video), 4 subjects
Lalys (2011) [22]	DTW	Yes	Surgical phase recognition	Cataract surgery, 20 videos
Blum (2010) [23]	CCA, HMM	Yes	Surgical phase recognition	Laparoscopic surgery, 10 videos
Lin (2009) [24]	HMM	Yes	Skill classification but not on individual OSATS criteria	RMIS (both kinematic and video data from robotic surgery), 6 subjects

2.2 Surgical Activity Recognition

The problem of surgical activity recognition has been of interest to many researchers. Several methods have been proposed to develop algorithms that automatically recognize the phase of a surgery. Some of the initial works focused on very low level gesture recognition in RMIS training [25, 26, 13, 27], while more recently, many works have focused on recognizing high level phases in surgeries [17, 28, 29, 30].

Several RMIS works have used Hidden Markov Models (HMMs) to represent the surgical motion flow. The motivation for HMMs and gesture based analysis is derived from speech recognition techniques and the goal is to develop a language of surgery where a surgical task can be modeled as a sequence of predefined gestures (also known as surgemes analogous to phonemes in speech recognition). [12] proposed a combined Markov/semi-Markov conditional random field (MsM-CRF) model for gesture segmentation and recognition for RMIS. [19] and [20] employed both kinematic and video data while using linear dynamical systems (LDS) and bag-of-features (BoF) for surgical gesture (surgeme) classification in RMIS surgery. [18] developed a method to capture long-range state transitions between actions by using higher-order temporal relationships using a variation of the Skip-Chain Conditional Random Field. Some more recent works have presented unsupervised methods to identify similar low-level trajectories with strong alignment to human labels [31, 32].

Unlike most of the RMIS based work described above where the focus was on recognizing low level gestures, multiple approaches have been presented to recognize high level surgical tasks in laparascopic surgeries[21, 17, 29, 30]. In [21], the authors presented a DTW and HMM based method for recognizing surgical phases using tool usage recordings as a multidimensional time series. [30] proposed a fully data-driven and real-time method for segmentation and recognition of surgical phases using a combination of video data and instrument usage signals. More recently, with the immense success of deep learn-

ing in image recognition fields, some works have proposed convolutional neural networks (CNN) based methods for surgical phase recognition. [17] collected a new dataset of 80 laparascopic cholecystecomies (Cholec80) and proposed *'EndoNet'*, a modified version of AlexNet, to simultaneously recognize surgical tools and phase. A few works have then tried to improve surgical tool and phase recognition using various deep learning models [28, 33, 34]. Outside of laparascopic domain, some works have also presented methods on recognizing surgical tasks in ENT [35] and cataract surgeries [36].

Most of the surgical activity recognition work described above has focused on low level gesture recognition in basic RMIS tasks. The few works presented for recognizing steps of clinical procedures mainly focused on laparoscopic surgeries with little to no work done on recognizing surgical steps in a robot-assisted clinical procedure.

2.3 Conclusion

To better guide our research forward, we can derive the following conclusions from the literature survey:

1. The majority of the work done on surgical skills assessment has been focused on basic RMIS training. Very few papers have presented methods for OSATS based assessment in general surgical training. Moreover, most of such works use small datasets.

2. Most of the real surgery phase recognition work has been done on laparoscopic procedures - little to no research has been done on recognizing surgical steps in clinical robot-assisted surgeries.

3. There is a lack of work done on providing automated feedback to surgeons for robot-assisted surgeries in a clinical setting. Most of the feedback given in such cases is limited to gross measures across the entire procedure despite the performance of particular tasks being largely responsible for undesirable outcomes.

CHAPTER 3

SURGICAL SKILL ASSESSMENT IN BASIC TRAINING

Surgical trainees are required to acquire specific skills during the course of their residency before performing real surgeries. The first skills that trainees need to master are those of basic suturing and knot tying. These skills form a base for any future skills that surgical trainees need to acquire. Therefore, mastering the art of suturing and knot tying is very essential in the career of any surgeon. However, due to the packed schedule of supervising surgeons, trainees usually do not get frequent feedback that is necessary for their learning. Moreover, the manual assessment by experts can be subjective and prone to errors. Objective Structured Assessment of Technical Skills (OSATS) is adopted in most medical schools as a standard to assess surgical residents [1] (see Table 1.1 for details on OSATS grading scheme). While adopting OSATS grading system reduces the subjectivity of assessment to some extent, the grading itself can take up lot of time of the generally few expert surgeons available. In this chapter, we present a framework for automated OSATS based surgical skills assessment for basic surgical tasks of suturing and knot tying using video and accelerometer data.

3.1 Methodology

Figure 3.1 shows the flow diagram for processing video and accelerometer data for surgical skills assessment. The videos are initially preprocessed and converted into a multi-dimensional time-series, whereas, the accelerometer data is first aligned with the video data before further processing. We will now go into more details of the different parts in the pipeline below.

Chapter references: [37, 38, 39]

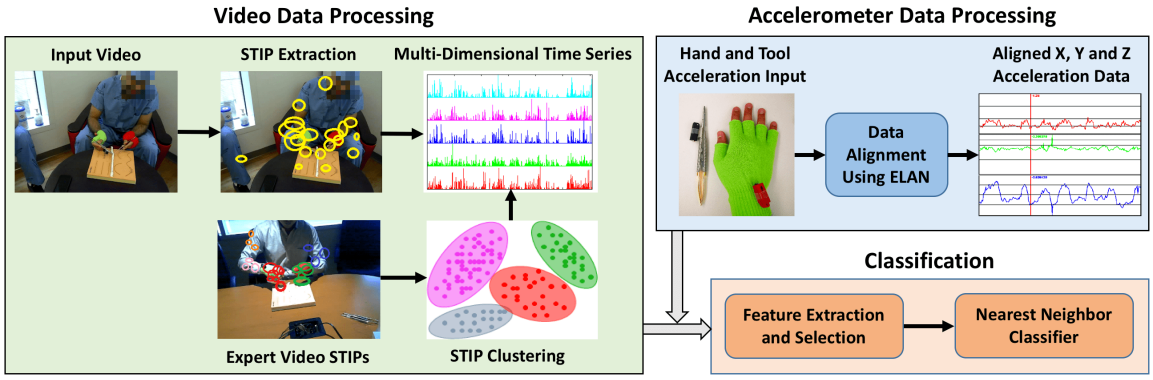

Figure 3.1: Flow diagram for processing the video and accelerometer data.

3.1.1 Video/Accelerometer Data Processing

In order to extract motion information from video data, we use Spatio-Temporal Interest Points (STIPs) [40] proposed by Laptev. Let V be the set containing all the videos in our dataset. Then, for all $v \in V$, a Harris3D detector is used to compute the spatio-temporal second-moment matrix μ at each video point given by

$$\mu = g(.;\sigma^2,\tau^2) * \begin{pmatrix} L_x^2 & L_x & L_xL_t \\ L_xL_y & L_y^2 & L_yL_t \\ L_xL_t & L_yL_t & L_t^2 \end{pmatrix} \tag{3.1}$$

where $g(.;\sigma^2,\tau^2)$ is a 3D Gaussian smoothing kernel with a spatial scale σ and a temporal scale τ. $L_{x,y,t}$ are gradient functions along the x, y and t domains. The final position of the STIPs is then calculated by finding the local maxima of the Harris corner function given by

$$H = \det(\mu) - \omega(\text{trace}(\mu))^3 \tag{3.2}$$

Laptev's STIP implementation [41] was used with default parameters and sparse feature detection mode for different spatio-temporal scales with ω set to be 0.005. Histogram of of Optical Flow (HOF) and Histogram of Oriented Gradients (HOG) are then computed on a three-dimensional video patch in the neighborhood of each detected STIP. A 4-bin

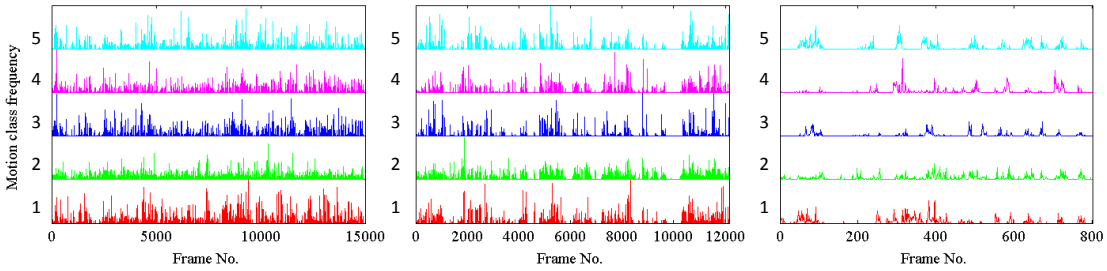

Figure 3.2: Motion class time series samples using $K = 5$ for a novice (left), an intermediate (center) and an expert (right) surgeon.

HOG and a 5-bin HOF descriptor is calculated resulting in 72-dimensional HOG vector and a 90-dimensional HOF vector. The final feature vector for each STIP is obtained by concatenating HOG and HOF vectors resulting in a 162-dimensional vector.

Once the STIPs for all videos are extracted, *'motion classes'* are learned by using k-means clustering on STIPs from two expert videos. Expert STIPs are used since they are more distinct and uncluttered as compared to non-experts. Therefore, expert motions provide exemplary templates for the surgical task to be evaluated. The STIPs from experts are clustered using k-means for different number of clusters 'c'. The learned clusters can be thought of as representing of the number of moving parts in the video. The expert clusters are then used to transform the remaining videos in the data set into a multi-dimensional time series. This is done by assigning each STIP in every frame of the video to one of the 'c' learned clusters using minimum Mahalanobis distance from the cluster distribution. This results in a time series $T \in \Re^{K \times N}$ representing each video, where K represents the dimension of the time series (equivalent to the number of clusters used in k-means) and N is the number of frames of the video. Figure 3.2 shows some sample motion class time series for a beginner, intermediate and an expert using $K = 5$.

The accelerometer data collected was already in a multi-dimensional time series format. Each data recording for an individual accelerometer resulted in a time series $T \in \Re^{3 \times N}$, where the rows denote the 3 acceleration values (x,y and z).

3.1.2 Feature Extraction

The difference in motion predictability and repeatability of surgeons with varying skills levels can potentially be used to assess the basic surgical skills. An expert will have more predictable hand motion while a beginner will exhibit erratic and irregular patterns. Therefore, we propose to use frequency based (DCT and DFT) and entropy based (ApEn and XApEn) features for extracting predictability and repeatability in time series data for skill assessment. Details of the different features used are given below

Discrete Fourier Transform: Discrete Fourier Transform (DFT) is used to convert data from time domain into frequency domain and has been extensively used for many application across several domains. For our time series $X \in \Re^{K \times N}$, we calculate the frequency coefficients for each dimension independently and concatenate them to form the frequency matrix $Q \in \Re^{K \times N}$ [37]. The i^{th} row in the frequency matrix Q, $Q(i)$ is calculated by

$$Q(i) = \theta X(i)' \tag{3.3}$$

where $X(i)$ is the i^{th} dimension of the time series X. θ is an $N \times N$ matrix and $\theta(m, n)$ is given by

$$\theta(m, n) = \exp(-j2\pi \frac{mn}{N}), \tag{3.4}$$

where $\{m, n\} \in [0, 1, \ldots, N-1]$. Once the matrix Q is calculated, the higher frequency terms are removed in order to eliminate noise. This results in a reduced matrix $\hat{Q} \in \Re^{K \times F}$ where F denotes the highest frequency component used from each dimension of the time series X. This can also be thought of as low-pass filtering of the time series. The elements of $\hat{Q}$ are then concatenated to form a final feature vector of KF dimensions.

Discrete Cosine Transform: Discrete Cosine Transform (DCT) is also a transformation of data from time domain to frequency just like DFT. However, DCT only uses cosine functions instead of both sines and cosines. This results in the DCT coefficients being real

as opposed to DFT where the coefficients can be complex. Similar to DFT, the i^{th} row of the frequency matrix $Q \in \Re^{K \times N}$ is also calculated using equation 3.3 [37] but the θ matrix is given by

$$\theta(0, n) = \sqrt{\frac{1}{N}}, \tag{3.5}$$

$$\theta(m, n) = \sqrt{\frac{2}{N}} cos(\frac{\pi(2n+1)m}{2N}), \tag{3.6}$$

where $\{m, n\} \in [0, 1, \ldots, N-1]$. Similar to DFT, the matrix Q is reduced to $\hat{Q} \in \Re^{K \times F}$ and a final KF-dimensional feature vector is obtained.

Approximate Entropy: Approximate entropy is a measure of regularity in time series data initially proposed in [42]. A more predictable time series would have a low approximate entropy value whereas an irregular time series would have a higher entropy. For a one-dimensional time series, the approximate entropy *ApEn* is dependent on three parameters: embedding dimension (m), radius (r) and time delay (τ). The embedding dimension (m) represents the length of the series which is being checked for repeatability, the radius (r) is used for local probabilities estimation and time delay (τ) is selected in order to make the components of the embedding vector independent. For a given time series $T \in \Re^N$, we form a sequence of embedding vectors $x(1), x(2), \ldots, x(N-m+1)$, where $x(i)$ is given by $x(i) = [T_i, T_{i+\tau}, \ldots, T_{i+(m-1)\tau}]$, for $1 \leq i \leq N-(m-1)\tau$. Then, for each embedding vector $x(i)$, the frequency of repeatable patterns $C_i^m(r)$ is calculated by

$$C_i^m(r) = \frac{1}{N-(m-1)\tau} \sum_j H(r - dist(x(i), x(j))) \tag{3.7}$$

where H is a Heaviside step functions and

$dist(x(i), x(j)) = \max(|T(i+(k-1)\tau) - T(j+(k-1)\tau)|)$ for $k \in [1, 2, \ldots, m]$. The

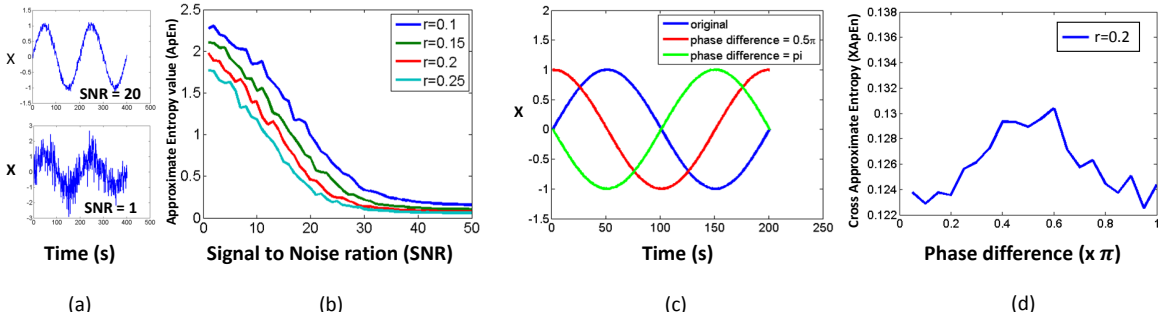

Figure 3.3: (a) Sample sine waves with different SNR. (b) Variation of approximate entropy *(ApEn)* with respect to SNR (c) Sample sine waves with different phases (d) Variation of cross approximate entropy *(XApEn)* with respect to phase difference between signals

conditional frequency estimates are calculated by

$$\Omega^m(r) = \frac{1}{N-(m-1)\tau} \sum_{i=1}^{N-(m-1)\tau} ln(C_i^m(r) \tag{3.8}$$

$\Omega(r)$ is then used to calculate the approximate entropy for the time series $T \in \Re^N$ as $ApEn(m,r,\tau) = \Omega^m(r) - \Omega^{m+1}(r)$.

In order to show how *ApEn* varies for signals with different predictability, we generate a set of sinusoids V. A pure sine wave without any noise can be considered as completely predictable since it has a fixed repeating pattern. However, adding noise to the same function would make it less predictable. We induce white Gaussian noise into our set of sinusoids V to vary the signal-to-noise (SNR) of the set of signals. The range of SNR in the set V was kept from 1 to 50. Figure 3.3(a) shows some sample sinusoidal waves in the set V with different SNR. Figure 3.3(b) shows the variation of *ApEn* with varying SNR and radius. As expected, we can see that the higher the SNR (lesser noise), the lower the value of *ApEn* gets for any value of r.

Cross Approximate Entropy: Cross approximate entropy *(XApEn)* is a measure of asynchrony between two time series [43]. For two given time series $[T,S] \in \Re^N$, the embedding vectors are defined as $x_1(i) = [T_i, T_{i+\tau}, \ldots, T_{i+(m-1)\tau}]$ and $x_2(i) = [S_i, S_{i+\tau}, \ldots, S_{i+(m-1)\tau}]$, for $1 \leq i \leq N-(m-1)\tau$. The frequency of repeatable patterns $C_i^m(r)(T||S)$ for the

embedding vectors $x_1(i)$ and $x_2(i)$ is then calculated by

$$C_i^m(r)(T||S) = \frac{1}{N-(m-1)\tau} \sum_j H(r - dist(x_1(i), x_2(j))) \tag{3.9}$$

$\Omega^m(r)$ is then calculated using

$$\Omega^m(r) = \frac{1}{N-(m-1)\tau} \sum_{i=1}^{N-(m-1)\tau} ln(C_i^m(r)(T||S)) \tag{3.10}$$

This is then used to finally calculate the cross approximate entropy between the two time series by

$XApEn(m, r, \tau) = \Omega^m(r)(T||S) - \Omega^{m+1}(r)(T||S)$.

Similar to *ApEn*, we generate a set of sinusoids W to show the variation of *XApEn* for varying synchrony between different signals. The set W consists of sinusoids with the same SNR but with phase varying from 0 to π. Figure 3.3(c) shows some sample of sinusoids in this set. Figure 3.3(d) shows how the value of *XApEn* varies when the phase difference between the signals varies. We can see that the value of *XApEn* reaches a max at about 0.5π and then reduces back to 0 at π phase difference. It is important to note that two sinusoids with a phase difference of π are completely out of phase but in perfect synchrony. This is because if one increases the other decreases with the same rate. This should result in a very low *XApEn* value which we observe in Figure 3.3(d) as well.

Surgical motions in suturing and knot tying tasks are inherently repetitive in nature. The repetitiveness of motion can be encoded using frequency features. However, frequency features would not be able to capture the sudden movements or jerks in motion that define the competency of a surgeon. They do not quantify the orderliness or predictability of patterns. On the other hand, approximate entropy represents the likelihood of occurrence of similar patterns of observations. A time series containing many repetitive patterns has lower approximate entropy and is more predictable.Therefore, using *ApEn* features can potentially capture repetitiveness along with more finer details crucial for skills assessment.

Moreover, in surgical motions, it is also important for surgeons to move their hands and tools in a smooth motion together. We think that *XApEn* features can potentially capture information on how synchronized the surgeon's hands and tools are with each other. We use both the entropy based features described above to encode surgical motion predictability for our analysis.

3.1.3 Classification

After extracting the features described above, we use Sequential forward selection (SFS) [44] to reduce the dimensionality of the features. Finally, a Nearest-Neighbor (NN) classifier is used for classification.

3.2 Experimental Evaluation

3.2.1 Data Set

Our data set consists of video and accelerometer data for evaluating the performance of proposed and previous state-of-the-art features for skill assessment. We use the surgical skills dataset from [37] for direct comparisons. This dataset had 18 participants. We augmented this dataset with additional 23 participants to a total of 41 participants consisting of surgical residents and nurse practitioners, essentially doubling the data set from previous studies. In suturing, the participants were asked to perform a "*running suture*" using an instrument (needle holder) for a specified amount of time, resulting in varied number of sutures completed. For knot tying, the participants were asked to tie knots for a given time using their hands only (without any instruments). In this data set, each participant undertook two instances each of suturing and knot tying tasks. For each instance, video data was captured at 30 frames per second at a resolution of 640×480 using a standard RGB camera. We captured a fixed number of frames for each surgical task: 4000 for suturing and 1000 for knot tying. Each video was captured in different lighting conditions and from varying camera angles to make the data set invariant to lighting and viewing angle. Figure

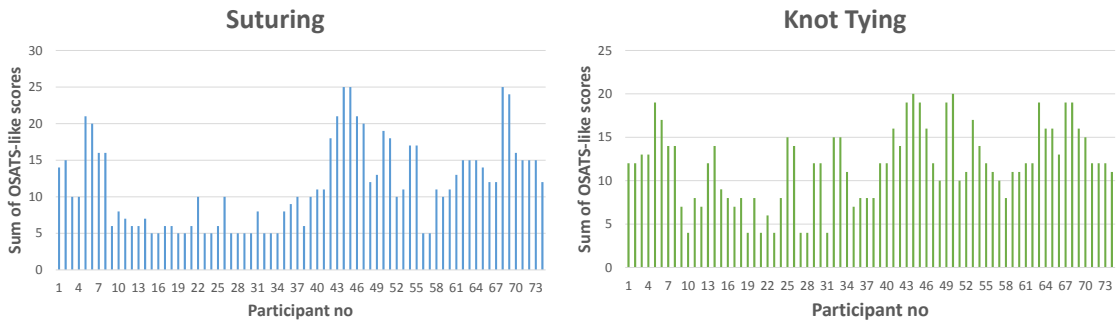

Figure 3.4: OSATS score distribution for both tasks in the dataset. For this plot, the individual scores for each criteria were summed for each participant.

3.5 shows some sample frames from the videos. Due to acquisition errors, some videos had to be excluded from the data set resulting in 74 videos (from 38 participants) for each surgical task.

The acceleration data was captured using Axivity WAX9[1] sensors. Two accelerometers were used for each surgical task. For knot tying, one accelerometer was attached to each hand wrist whereas for suturing, one accelerometer was attached to the dominant hand wrist and one to the needle-holder. This was done because for suturing, there was very little movement of the non-dominant hand and would not contribute much. On the other hand, needle holder is the main instrument used for suturing. Hence we capture the motion of the dominant hand and the needle holder for suturing. The data captured consisted of x, y and z acceleration values resulting in a 3-dimensional time series for each accelerometer. At the start of each instance, all participants were asked to rapidly shake the hands/instruments with the accelerometers to get the synchronization waveform that is used to align the starting point of acceleration data with the video using the ELAN software [45] (a snapshot shown in figure 3.5). The accelerometer data had some additional noise as the accelerometers were not being attached properly, resulting in unwanted jerks. For some cases, the accelerometer even fell off during a session and had to be reattached. All such samples were removed from the data set resulting in a final 54 acceleration data samples for

[1] https://axivity.com/downloads/wax9

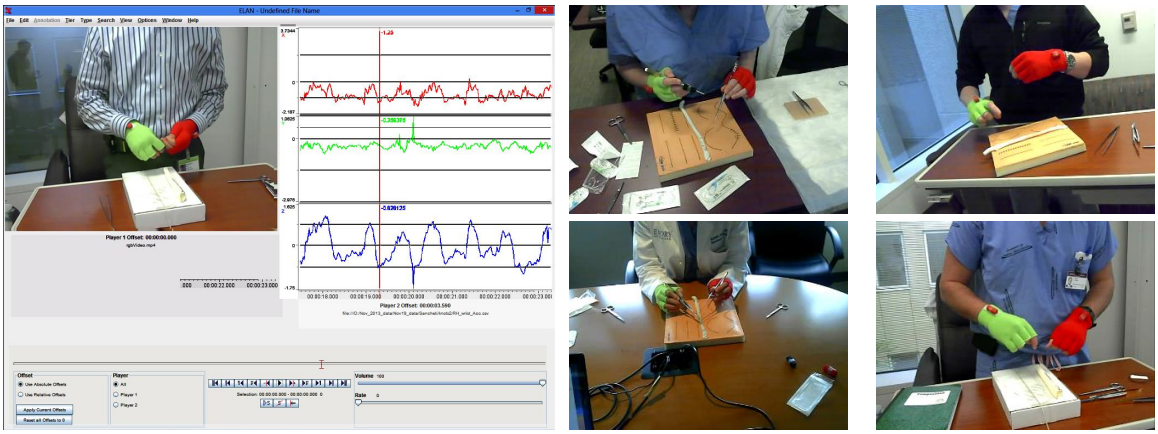

Figure 3.5: Image on left shows a screenshot from ELAN software for synchronization of video and accelerometer data. Middle column and right most columns show sample frames for suturing and knot tying, respectively. The accelerometers can also be seen placed on the wrists and the needle-holder

knot tying (from 30 participants) and 62 for suturing (from 33 participants). The average length with standard deviations of the acceleration data was 8434 ± 2030 for suturing and 1919 ± 507 for knot tying.

In order to generate the ground truth skill levels, we asked an expert to watch the videos and give OSATS scores (on a scale of 1 to 5) for each participant. The scores were then divided into three categories: beginner ($score = [1, 2]$), intermediate ($score = 3$) and expert ($score = [4, 5]$). A complete class distribution for video and accelerometer data is given in Table 3.1. We also show the distribution of the sum of OSATS scores in Figure 3.4 for both tasks. Please note that we only use the OSATS criteria being used in our partner hospital for actual assessment. For example, RT and IH were not used for knot tying since there is no direct tissue contact with no instrument being used. Scores for OP in suturing and KP in both tasks, were not available.

In order to generate the ground truth skill levels, we asked an expert to watch the videos and give OSATS scores (on a scale of 1 to 5) for each participant. The scores were then divided into three categories: beginner ($score = [1, 2]$), intermediate ($score = 3$) and expert ($score = [4, 5]$).

Table 3.1: Skill class distribution for each of the OSATS criteria (RT: Respect for Tissue, TM: Time and Motion, IH: Instrument Handling, SH: Suture Handling, FO: Flow of Operation, OP: Overall Performance). Each cell contains two values $V : A$, where V = No. of samples for video data, A = No. of samples for acceleration data.

	Suturing					**Knot Tying**			
	RT	**TM**	**IH**	**SH**	**FO**	**TM**	**SH**	**FO**	**OP**
Beginner	38 : 28	46 : 34	47 : 35	47 : 35	45 : 33	27 : 18	27 : 19	22 : 15	23 : 15
Intermediate	22 : 20	15 : 15	13 : 13	17 : 17	18 : 18	22 : 17	28 : 21	28 : 22	28 : 22
Expert	14 : 14	13 : 13	14 : 14	10 : 10	11 : 11	25 : 19	19 : 14	24 : 17	23 : 17

3.2.2 Parameter Selection

There are multiple parameters that we need to find optimal values for in different parts of our pipeline. First parameter that we tuned was the dimension of time series data to be used from videos i.e. the number of motion classes. We used $K \in [2, 3, \ldots, 10, 12, \ldots, 20]$ for k-means clustering to learn motion classes (the number of time series dimensions used) for analysis of video data. Each feature used had different optimal values of K and are given in Table 3.2.

For frequency based methods described, the only parameter that needs to be selected empirically is F which is the highest frequency component selected from each dimension of the time series (or the cutoff frequency in the low pass filter). Therefore, we calculate the classification accuracy for $F \in [25, 50, 100, 200, 500]$. Average accuracies were evaluated over all OSATS criteria and $F = 50$ achieved the best performance. We will maintain $F = 50$ for our evaluation and results comparison.

As described in the previous section, entropy based features are dependent on some parameters which need to be specified. These are the embedding dimension (m), time delay (τ) and the radius (r). In order to differentiate time series data on the basis of regularity, radius (r) needs to be equal to $r_{coeff} \times std$, where r_{coeff} can range from 0.1 to 0.25 and std denotes the standard deviation of the time series. For the embedding dimension, $m = 1$ and $m = 2$ both work equally well according to [42]. The time delay τ essentially represents the factor by which the input data is down sampled for further calculations.

3.2.3 Evaluation Metrics

Different metrics were used to compare performances of various features on our data set. For video, we calculate the average classification accuracy over all OSATS criteria for different features for all values of K in order to find the optimum number of clusters for each feature type. The average accuracy $\hat{A}_k$ is calculated using $\hat{A}_k = \frac{1}{O} \sum_{OSATS} A_K$, where A_K is the accuracy using K clusters for a specific OSATS criteria, and O represents the total number of applicable OSATS criteria for that task. For accelerometer data, we evaluate the different features for both the accelerometers attached for each task; wrist and needle-holder for suturing and hand wrists for knot tying. Accuracies are averaged over all OSATS criteria for accelerometer data as well.

We also calculate the class wise precision and recall values as $precision = \frac{tp}{tp+fp}$ and $recall = \frac{tp}{tp+fn}$, where tp is true positive, fp is false positive and fn denotes the false negatives for the corresponding class. Again, the per-class precision and recall values are averaged over all OSATS criteria for a more compact representation.

3.3 Results

The features and evaluation metrics described in the previous section were evaluated on video and accelerometer data for suturing and knot tying tasks for all applicable OSATS

Table 3.2: Highest average classification accuracies with standard deviations for different techniques using multi-modality data. For video data, K corresponding to highest accuracy is also shown.

	Video		Accelerometer	
	Suturing	Knot Tying	Suturing	Knot Tying
SMT	78.9 ± 5.7 (K=3)	61.1 ± 2.3 (K=10)	72.9 ± 4.5	72.7 ± 5.3
DCT	91.9 ± 3.4 (K=9)	89.5 ± 2.8 (K=9)	84.5 ± 4.9	83.3 ± 2.1
DFT	92.4 ± 3.7 (K=7)	86.8 ± 2.8 (K=10)	85.5 ± 3.0	**84.7 ± 4.1**
ApEn	93.7 ± 2.2 (K=20)	89.2 ± 5.3 (K=20)	80.3 ± 2.1	75.0 ± 6.5
XApEn	91.4 ± 3.0 (K=16)	90.9 ± 4.3 (K=20)	81.0 ± 4.0	66.2 ± 4.1
ApEn+XApEn	**95.1 ± 3.1 (K=16)**	**92.2 ± 3.0 (K=14)**	**86.8 ± 4.5**	78.7 ± 5.8

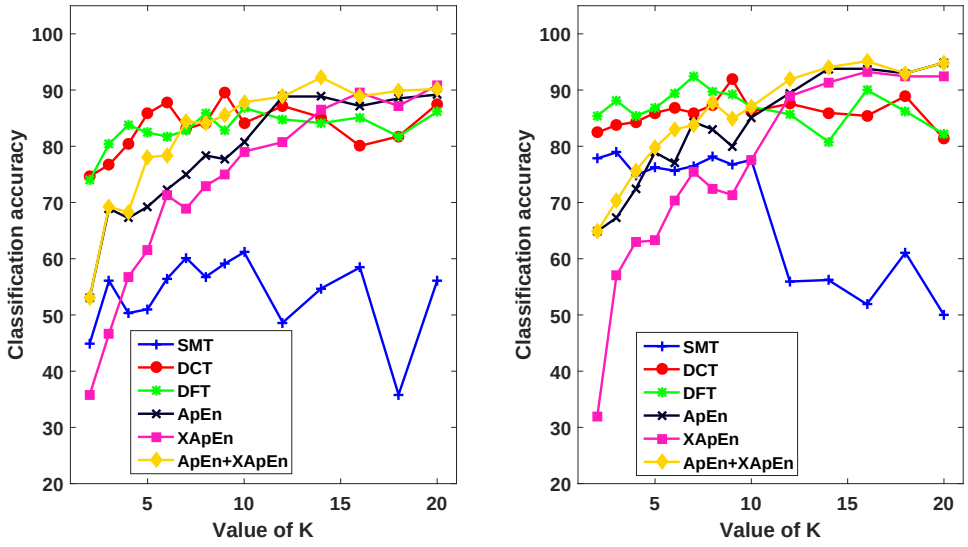

Figure 3.6: Average classification accuracy ($\hat{A}_k$) versus K (number of dimensions of time series) for video data. (Best viewed in color)

Table 3.3: Per class average precision and recall values over all OSATS criteria with standard deviations using video data corresponding to Table 3.2. The values in each cell are in the format *Precision* | *Recall*.

	Suturing			Knot Tying		
	Beginner	Intermediate	Expert	Beginner	Intermediate	Expert
SMT	89.0±5.7 | 82.3±4.5	66.2±11.8 | 73.2±15.5	60.9±14.7 | 72.6±18.1	68.4±5.6 | 65.5±2.0	51.8±9.4 | 58.4±4.7	63.7±6.9 | 59.4±11.1
DCT	97.3±2.8 | 94.0±1.6	79.2±14.6 | 90.7±7.6	86.0±10.1 | 85.4±10.6	86.5±7.4 | 92.2±7.0	88.3±5.5 | **92.2±5.7**	**93.1±5.9** | 85.1±2.1
DFT	96.5±2.8 | 94.0±2.5	82.1±11.8 | 90.7±8.9	91.0±9.3 | 89.3±5.2	88.1±7.9 | 85.6±4.9	**91.5±7.3** | 85.3±5.4	79.7±9.9 | 90.8±7.7
ApEn	97.6±1.9 | **96.3±2.9**	86.7±8.4 | 90.1±3.0	**94.6±5.0** | 95.3±4.4	91.1±4.4 | 90.0±5.3	86.8±4.5 | 84.8±7.8	89.5±8.1 | **93.8±3.7**
XApEn	97.6±2.4 | 92.8±3.6	80.6±9.0 | **92.6±6.9**	93.8±6.2 | 96.6±4.6	91.6±4.1 | **94.2±3.9**	88.6±3.3 | 87.9±7.6	91.9±9.4 | 91.8±7.5
ApEn+XApEn	**98.1±2.2** | 95.2±3.2	**92.4±7.0** | 92.2±5.2	89.3±8.6 | **100.0±0.0**	**95.0±3.9** | 93.0±6.8	89.7±5.1 | 91.4±3.1	91.6±8.4 | 93.7±6.3

Table 3.4: Per class average precision and recall values over all OSATS criteria with standard deviations using accelerometer data corresponding to Table 3.2. The values in each cell are in the format *Precision* | *Recall*.

	Suturing			Knot Tying		
	Beginner	Intermediate	Expert	Beginner	Intermediate	Expert
SMT	82.3±4.1 | 79.0±5.0	60.7±7.9 | 69.0±7.8	63.9±11.0 | 61.6±13.4	54.8±8.4 | 75.4±11.5	81.0±7.2 | 66.9±3.2	80.4±3.5 | 80.6±9.6
DCT	**95.8±4.4** | 83.1±5.5	80.2±7.5 | **88.0±5.2**	60.5±7.9 | 84.7±15.0	83.6±9.1 | 79.7±9.7	85.3±7.6 | **84.7±3.3**	80.4±3.5 | 87.3±9.3
DFT	94.2±5.5 | **88.7±3.5**	**82.1±7.5** | 82.8±7.3	67.2±12.7 | 81.9±11.0	**84.9±3.8** | **87.8±6.8**	**88.1±5.7** | 78.3±2.0	80.7±6.9 | **91.4±3.8**
ApEn	91.9±4.2 | 82.5±3.0	64.1±10.0 | 76.1±10.0	69.1±6.0 | 76.4±6.0	74.0±15.1 | 69.0±10.3	67.7±6.6 | 76.5±6.2	82.7±10.7 | 80.9±10.5
XApEn	90.7±5.5 | 82.9±6.7	73.0±17.2 | 78.2±9.5	61.9±15.5 | 82.9±7.4	54.3±7.4 | 70.6±15.4	70.4±5.3 | 63.6±6.8	72.7±10.7 | 67.7±4.2
ApEn+XApEn	93.9±2.1 | 86.2±6.8	75.5±13.6 | 86.4±5.9	**81.7±14.0** | **93.2±7.5**	77.7±8.9 | 75.8±10.7	72.3±10.0 | 81.3±1.8	**86.0±9.8** | 79.3±8.3

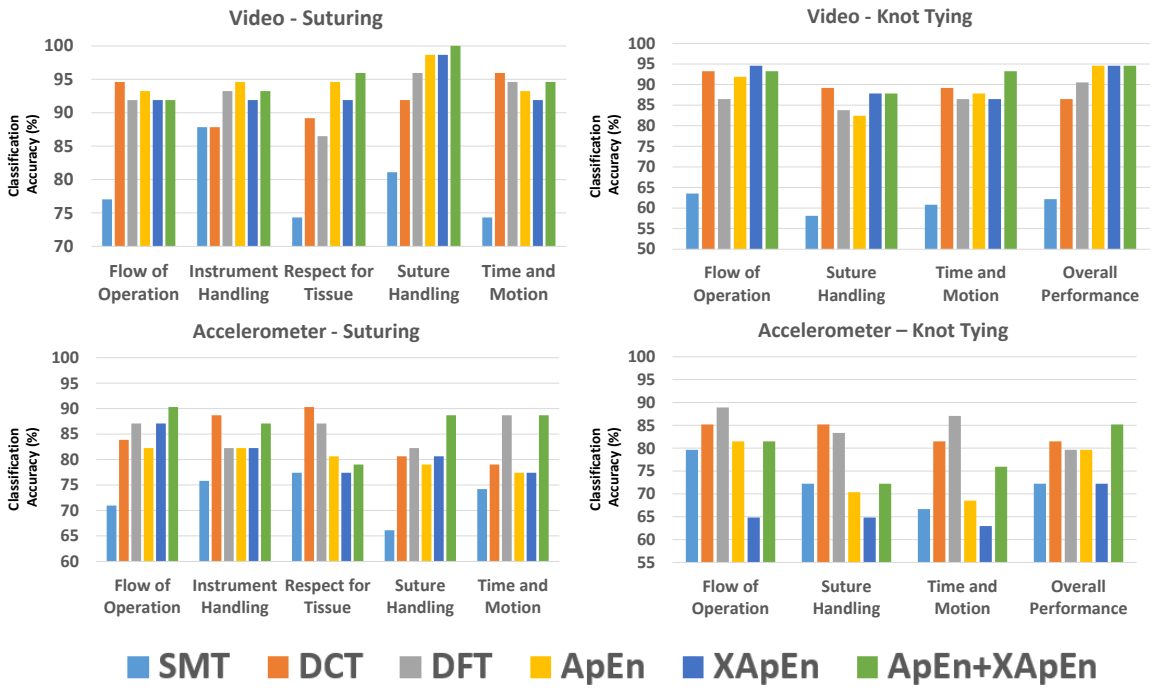

Figure 3.7: Individual OSATS criteria results for video and accelerometer data. For each feature, the optimal value of K (as indicated in Table 3.2) was used. (Best viewed in color)

criteria. Figure 3.6 shows the comparison of different features for suturing and knot tying tasks using video data while using different values of K. Figure 3.8 shows the average classification results achieved using accelerometer data. The highest average accuracy and the corresponding standard deviations achieved for different techniques are given in Table 3.2. Along with highest average accuracies, we also show the results for individual OSATS criteria using optimal K for each feature type (as indicated in Table 3.2) in Figure 3.7. The per-class precision and recall values corresponding to accuracies given in Table 3.2 are given in Tables 3.3 and 3.4.

In order to check the statistical significance of the presented results in Table 3.2, we conducted McNemar's test [46]. The best performing feature for each modality and surgical task was compared with the rest of the features. For comparing performance of different classifiers, a p-value < 0.05 indicates that the difference in classification accuracies is statistically significant. Table 3.5 shows the p-values achieved conducting the McNemar's

Table 3.5: McNemar's test of statistical significance for results presented in Table 3.2. For each column, the highest performing feature (denoted by *"HPF"*) was compared with all other features to check if the higher accuracy achieved is statistically significant by evaluating the p-value. For example, in the first column, ApEn+XApEn performance was compared to rest. The improvement in accuracy is statistically significant if p-value$<$0.05.

	Video		Accelerometer	
	Suturing	Knot Tying	Suturing	Knot Tying
SMT	$<$0.01	$<$0.01	$<$0.01	$<$0.01
DCT	$<$0.01	$<$0.01	$<$0.05	$<$0.05
DFT	$<$0.01	$<$0.01	$<$0.05	HPF
ApEn	$>$0.05	$<$0.01	$<$0.01	$<$0.01
XApEn	$<$0.01	$<$0.05	$<$0.05	$<$0.01
ApEn+XApEn	HPF	HPF	HPF	$<$0.01

test. It can be observed that the improvement in average classification accuracy by the highest performing feature for each column is statistically significant for almost all cases. This shows that the improvements achieved by the proposed entropy based features, when using video data for both tasks and using accelerometer data for suturing, is statistically significant.

We also perform experiments to compare how an early fusion of video and accelerometer data performs for frequency (DCT and DFT) and top performing entropy features (*ApEn+XApEn*). The features are fused via concatenation. Since some of the accelerometer data had to be excluded (as described in Section 4), we only use videos for which the corresponding accelerometer data is available i.e 54 for knot tying and 62 for suturing. Tables 3.6 and 3.7 show the average accuracies (over all OSATS criteria) with standard deviations using different modalities for suturing and knot tying, respectively.

Lastly, for a more thorough comparison, we perform another experiment using harder cross validation schemes. We again compare *ApEn+XApEn* with DCT and DFT. For this analysis, we use the Video+Acceleration data for each feature type. Figure 3.9 shows the average accuracies with standard deviation over all OSATS criteria for 2, 5, and 10 fold

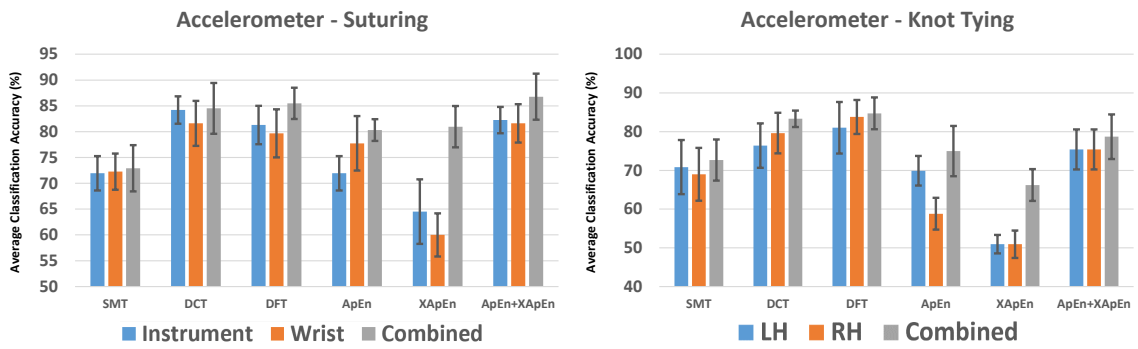

Figure 3.8: Average classification accuracies with standard deviations for accelerometer data using individual and combination of the two accelerometers. (Best viewed in color)

Table 3.6: Average accuracies with standard deviations for corresponding feature types using different data modalities for suturing task. Highest performance across all modalities and feature types is shown in bold

	Video	Accelerometer	Video+Accelerometer
DCT	90.6 ± 3.1	84.5 ± 4.9	86.8 ± 7.7
DFT	87.1 ± 1.1	85.5 ± 3.0	86.1 ± 2.1
ApEn+XApEn	$\mathbf{93.9 \pm 3.7}$	86.8 ± 4.5	93.2 ± 6.6

cross validation schemes. Tables 3.8 and 3.9 show results for *'hold-out'* cross validation schemes for suturing and knot tying, respectively. For hold-out validation scheme, $h\%$ of the data was kept as testing data (corresponding to each column in the tables) while the remaining $(100 - h)\%$ was used for training. Within the training data, 10% was used as validation set. Both validation and testing accuracies are given in Tables 3.8 and 3.9. We do not show training accuracy since that will always be 100% using a nearest-neighbor classifier (each point in the training data will be closest to itself, always).

Table 3.7: Average accuracies with standard deviations for corresponding feature types using different data modalities for knot tying task. Highest performance across all modalities and feature types is shown in bold

	Video	Accelerometer	Video+Accelerometer
DCT	91.7 ± 6.1	83.3 ± 2.1	83.8 ± 4.9
DFT	86.1 ± 1.9	84.7 ± 4.1	81.0 ± 5.5
ApEn+XApEn	90.3 ± 3.1	78.7 ± 5.8	$\mathbf{94.0 \pm 2.8}$

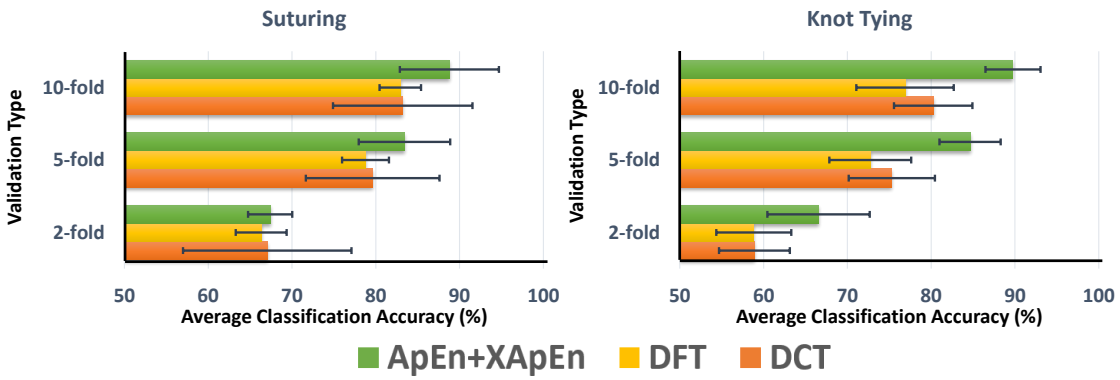

Figure 3.9: Average classification accuracy bars with standard deviations for different cross validation schemes by using Video+Accelerometer data. (Best viewed in color)

Table 3.8: Average validation and testing accuracies over all OSATS criteria with standard deviations using hold-out cross-validation for suturing with Video+Accelerometer data. The values in each cell are in the format *Validation Accuracy* | *Testing Accuracy*. Each column corresponds to the amount of data that was *left-out* for testing.

	Testing Set Percentage													
	80%		70%		60%		50%		40%		30%		20%	
DCT	50.3±8.7	51.8±7.6	55.8±8.4	57.7±9.0	**60.5±8.9**	61.9±9.2	64.3±9.6	66.3±9.3	69.1±9.5	71.8±8.7	72.5±8.7	75.4±8.9	76.3±8.8	79.3±8.3
DFT	**53.6±1.8**	**54.0±1.3**	**57.8±2.3**	**58.7±1.7**	**60.5±1.5**	**63.0±1.9**	65.0±1.9	67.3±2.2	69.3±2.2	71.6±2.4	73.1±1.9	75.3±2.4	76.2±2.4	79.0±2.4
ApEn+XApEn	51.6±2.8	51.5±2.3	56.0±3.0	56.9±3.2	59.9±3.5	62.7±4.0	**65.5±3.8**	**67.8±4.3**	**71.3±5.0**	**73.7±4.7**	**75.3±5.0**	**78.4±5.3**	**79.8±5.3**	**83.7±5.7**

Table 3.9: Average validation and testing accuracies over all OSATS criteria with standard deviations using hold-out cross-validation for knot tying with Video+Accelerometer data. The values in each cell are in the format *Validation Accuracy* | *Testing Accuracy*. Each column corresponds to the amount of data that was *left-out* for testing.

	Testing Set Percentage													
	80%		70%		60%		50%		40%		30%		20%	
DCT	42.4±3.7	45.0±3.6	48.5±5.0	50.6±3.9	53.9±5.1	54.9±4.4	57.9±4.0	60.4±4.5	63.2±4.5	65.0±4.1	67.6±4.4	70.2±4.6	71.8±4.8	75.9±5.0
DFT	45.7±3.2	45.4±4.5	50.9±5.4	50.4±5.0	52.7±4.7	54.9±4.9	57.8±4.9	58.7±5.3	6.3±5.0	63.9±5.3	65.6±5.7	68.1±5.1	70.2±5.5	73.3±6.1
ApEn+XApEn	**46.9±5.8**	**47.0±6.3**	**54.6±5.8**	**54.5±6.7**	**58.5±5.7**	**60.9±6.2**	**64.2±5.5**	**66.6±5.6**	**70.2±4.8**	**73.7±5.2**	**75.4±4.9**	**79.0±4.7**	**80.8±4.2**	**85.4±4.2**

3.4 Discussion

From the results presented in the previous section, we can see that entropy based features perform better for video data as compared to state-of-the-art techniques in terms of accuracy. For accelerometer data, entropy based features attain a higher accuracy for suturing but not for knot tying (Table 3.2). The reasons for this is mainly because entropy based features are dependent on the dimension of the time series used (can also be seen in Figure 3.6 for increasing values of K); the higher the dimension of time series being evaluated, the more information is captured especially for cross entropy (*XApEn*). In case of accelerometer data, we only have 3-axis acceleration values so entropy based features cannot capture enough information. However, entropy based features still have a higher accuracy for suturing task. From Tables 3.3 and 3.4, we can see that entropy based features perform well overall, however, there isn't a conclusive trend in terms of precision/recall values.

Comparing the performances of using individual or a combination of accelerometers from Figure 3.8, we can observe that the combination of data from both accelerometers performs better than individual accelerometers. However, these differences in the performance can potentially give us some valuable insights for skill assessment. For example, in suturing, instrument data works slightly better than wrist for most of the feature types. The reason for this could be that there is relatively more movement of the instrument in suturing as compared to the wrist. Therefore, more motion information would be available to differentiate between different skills. This information can help surgeons improve on their skills by focusing on their instrument motion a bit more.

Comparing results for individual modalities shows us that using video data performs much better than accelerometer for all feature types. This can be explained by the fact that accelerometers only capture the hands/needle-holder 3-D acceleration data whereas videos can be used to extract all motions (both hands, instruments etc.). From the results of our video and accelerometer features fusion experiment (Table 3.6 and Table 3.7), we can see

that combining video and accelerometer data deteriorates performance for DCT and DFT features as compared to video data. For *ApEn+XApEn*, the performance improves for knot tying but slightly decreases for suturing. Overall, the highest performance is achieved using *ApEn+XApEn* features for each task (shown in bold). Even while using harder cross validation schemes, the proposed *ApEn+XApEn* features outperform frequency based features for both tasks for most setups (Figure 3.9, Table 3.8, Table 3.9).

While out-performing the previously proposed features for skill assessment, *ApEn* and *XApEn* also have some limitations. Firstly, these features are somewhat dependent on the dimensionality of the time series data; they work better for high dimensional data, especially for *XApEn* (since it can capture more information). However, increasing dimensionality also leads to potential over-fitting. Moreover, *XApEn* is computationally expensive and can take a long time if extracted using CPU. However, this can be overcome if a GPU implementation is used. In [47], the authors showed that using GPU for extracting *XApEn* from a multi-dimensional time series can be more than 250x faster than using CPU. This would be particularly important for real time feedback.

Although, previously proposed frequency features perform reasonably well (especially for accelerometer data), we think that they perform well on repetitive surgical tasks like suturing and knot tying. We believe that the proposed entropy based features would perform better in other surgical procedures as well since they try to capture the irregularity in motion instead of just the repetitiveness. Specifically, it would be interesting to see how these features perform in the recently published JIGSAWS dataset [48] since it contains similar surgical tasks being performed on a da Vinci robot.

3.5 Summary

In this chapter, we presented a framework for automated surgical skills assessment for basic tasks of suturing and knot tying using video and accelerometer data. Overall, our analysis showed that videos are better for extracting skill relevant information as compared

to accelerometer. However, a fusion of video and accelerometer features can improve the performance. Also, the proposed combination of *ApEn* and *XApEn* performed best among all features. Having an automated system for surgical skills assessment can significantly improve the quality of surgical training. It would allow the surgical trainees to practice their basic skills a lot more with valuable feedback. Moreover, such a system could also help save expert surgeon's time that is spent on trainee assessment.

CHAPTER 4

SURGICAL SKILL ASSESSMENT IN RMIS TRAINING

With the rapidly increasing amount of Robot-Assisted Minimally Invasive Surgery (RMIS) around the world, the focus on robotic surgical training has increased tremendously. Typical robotic surgery training includes simulator based and dry lab exercises like suturing, knot tying and needle passing. Training on these tasks is crucial since it forms the base for advanced training procedures on pigs, cadavers and eventually, humans. However, the current assessment on such dry lab exercises is done manually by supervising surgeons which makes it prone to subjectivity and reduces the overall efficiency of training.

In this chapter, we will extend the work presented in the previous chapter to develop an automated framework for assessment of surgical skills in basic RMIS training and achieve state-of-the-art performance using frequency and entropy based features. As opposed to previous chapter's work where we used video and accelerometer data, here we will only use robot-kinematics data to assess skill.

4.1 Methodology

4.1.1 Skill Classification/Score Prediction

As opposed to previous proposed works on using different variants of HMMs [50] for skill assessment, we evaluate holistic features for predicting skill level using robot kinematics data. Figure 4.1 shows the proposed pipeline. For a given D-dimensional time series $S \in \Re^{D \times L}$, where L is the number of frames, we extract 4 different types of features: Sequential Motion Texture (SMT), Discrete Fourier Transform (DFT), Discrete Cosine

Chapter reference: [49]

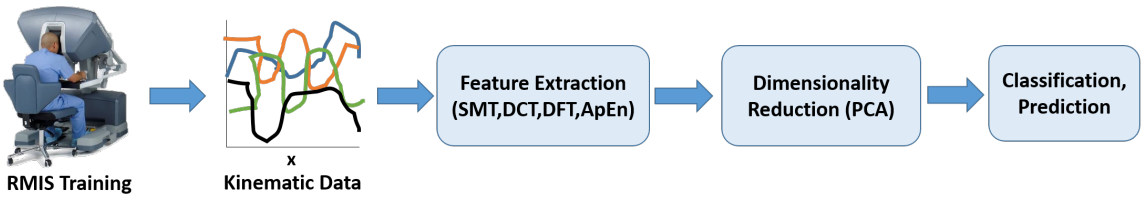

Figure 4.1: Flow diagram of the proposed framework for robotic surgical skills assessment.

Transform (DCT) and Approximate Entropy (ApEn). The dimensionality of the features is reduced using Principal Component Analysis (PCA) before classification/prediction. We give details of the feature types, fusion method and the prediction model below.

SMT: Sequential motion texture was implemented as presented in the original paper [5]. The time series is divided into N_w number of windows. A frame kernel matrix is calculated after which Gray Level Co-Occurence Matrices (GLCM) texture features (20 in total) are evaluated resulting in a feature vector $\phi_{SMT} \in \Re^{20N_w}$.

DCT/DFT: Frequency features were evaluated in a similar fashion as described in the previous chapter. We evaluate DCT and DFT coefficients for each dimension of the robot kinematics time series. This results in a matrix of frequency components $F \in \Re^{D \times L}$. The lowest Q components from each dimension are then concatenated together to make the final feature vector $\phi_{DCT/DFT} \in \Re^{DQ}$. Using low frequency features would eliminate any high frequency noise that could have resulted during data capture.

ApEn: Approximate entropy features were also extracted as presented in previous chapter. Evaluating ApEn for all dimensions of the time series data results in a feature vector

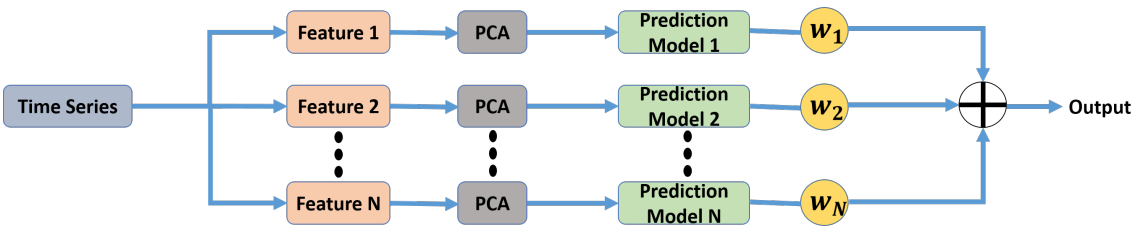

Figure 4.2: Weighted feature fusion for OSATS score and GRS prediction.

Figure 4.3: Sample frames from the 3 tasks in the JIGSAWS dataset [48].

$\phi_{ApEn} \in \Re^{DR}$, where R is the number of radius values used in evaluation per dimension.

Feature Fusion: A weighted feature fusion technique for skill prediction (as shown in Figure 4.2) is also used for classification/predictions. The outputs of different prediction models are combined to produce a skill score. We take our training time series data and evaluate each feature type to produce a training feature matrix $\phi_f \in \Re^{n \times D}$, where f corresponds to a the feature type used, n is the number of training samples and D is the dimensionality of the feature type. The output $y_f \in \Re^n$ corresponding to each ϕ_f is then evaluated using the prediction model. A matrix of outputs from different features $Y \in \Re^{n \times F}$ is generated by concatenating all the y_f, where F corresponds to total number of features used. Given the ground truth predictions $G \in \Re^n$, the optimal weights vector $w^* \in \Re^F$ is then evaluated by solving a simple least squares as $w^* = \underset{w}{\operatorname{argmin}} ||Yw - G||_2^2$. For a given test set, the output $\hat{y_{test}}$ is then calculated using $\hat{y_{test}} = Y_{test} w^*$.

Classification/Prediction: A simple nearest neighbor classifier is used for classification of skill levels. For exact score prediction,a linear support vector regression (SVR) model [51] is used.

4.2 Experimental Evaluation

Dataset: The proposed framework is evaluated on the publicly available JIGSAWS dataset [48]. This dataset is collected from a da Vinci Surgical System (dvSS) and consists of kinematics and video data from 8 participants for three robotic surgical tasks: Suturing, Knot Tying and Needle Passing. The video data is captured using the endoscopic camera

Table 4.1: Table showing optimal number of PCA components estimated. For prediction, the optimal value of the regularization parameter C is given within parentheses.

	SMT	DCT	DFT	ApEn
Classification	50	150	150	40
Prediction	10 (10^2)	1000 (10^{-6})	250 (10^{-6})	40 (10^4)

while the kinematic data consists of the joint information (e.g. Cartesian positions, linear and angular velocities, gripper angles, etc) from the different robot manipulators resulting in a 76-dimensional kinematic feature vector per frame. Figure 4.3 shows sample frames for each task. We only use kinematic data in its raw form without any preprocessing for our analysis and employ the standard LOSO (*leave-one-supertrial-out*) and LOUO (*leave-one-user-out*) cross validation setups. For LOSO, we leave one randomly selected trial from each surgeon out for testing and repeat this 20 times. For LOUO, we leave all trials from one surgeon out for testing. The dataset has ground truth skill labels of three categories: self-proclaimed, OSATS and global rating score (GRS). Self-proclaimed category has three skill levels (dependent on the amount of hours spent on the system) $-$ novice (< 10 hrs), intermediate ($10 - 100$ hrs) and expert (> 100 hrs). The OSATS scores are based on six criteria on a scale of 1-5 and are generated by an expert watching the videos while grading them. This is different from the original OSATS [3] (as described in introduction section) since it contains an extra criteria of suture handling (SH) and that none of the criteria are graded as Pass/Fail. The GRS is a sum of all individual OSATS scores.

Parameter estimation: There are different parameters that need to be tuned for the extraction of various features. We use the implementation of different features in their original forms as presented in the previous chapter. In SMT, we use number of windows $N_w = 10$ and evaluate Gray Level Co-Occurence Matrices (GLCM) texture features with 8 gray levels resulting in a 200-dimensional feature vector. For frequency features, we take the lowest 50 components ($Q = 50$) for each dimension of the time series and concatenate them resulting in a $50D$-dimensional feature vector, where D is the dimension of time series (76 in our case). In calculating approximate entropy ($ApEn$), we use radius

$r = [0.1, 0.13, 0.16, 0.19, 0.22, 0.25]$ resulting in a $6D$-dimensional feature vector. A value of 1 was used for both m and τ.

We use Principal Component Analysis (PCA) for dimensionality reduction before passing features onto the classifier or the regression model. This was done since a lower performance was observed using original feature dimensionality. In order to estimate the optimal number of PCA componenets D_{PCA}, we evaluate performance for D_{PCA} ranging from 10 to 3000 for all tasks for each feature type. The value of D_{PCA} corresponding to highest average performance accross all tasks was selected. For score predictions, we need to estimate an optimal value for the regularization parameter C in SVR. For each feature type, we evaluated the average correlation coefficient (over all OSATS) for $C \in [10^{-7}, 10^{-6}, \ldots, 10^{6}, 10^{7}]$ and selected the best performing value of C for evaluations. The optimal values of D_{PCA} and C are given in Table 4.1. Please note that all parameters were strictly tuned on the training data only for both validation setups. This includes the weights being estimated for the fusion of different prediction models.

4.3 Results and Discussion

We evaluate the proposed features for skill classification and OSATS based score prediction using the JIGSAWS dataset. For classification, we compare the performance of these features with previous HMM based state-of-the-art methods [50]. Table 4.2 shows results for self proclaimed skill level classification in the JIGSAWS dataset. As evident, using holistic features significantly out-perform previous approaches of using different variants of HMMs. Specifically, ApEn performs significantly better than all other methods. This is interesting to note since experts (with > 100 hrs of practice) would have smoother motions as compared to beginners (with < 10 hrs of practice) making their movements more 'predictable', and hence easily differentiated using ApEn features.

Table 4.3 shows the results for OSATS and global rating score predictions. We use spearman's correlation coefficient 'ρ' as an evaluation metric and check for statistical sig-

Table 4.2: Self proclaimed skill classification results

	Suturing		Knot Tying		Needle Passing	
	LOSO	LOUO	LOSO	LOUO	LOSO	LOUO
discrete-HMM	72.0	-	-	-	-	-
MFA-HMM	92.3	38.5	86.1	44.4	76.9	46.2
KSVD-HMM	97.4	59	94.4	58.3	96.2	26.9
SMT	99.0	35.3	99.6	32.3	99.9	57.1
DCT	**100**	64.7	99.7	54.8	99.9	35.7
DFT	**100**	64.7	**99.9**	51.6	99.9	46.4
ApEn	**100**	**88.2**	**99.9**	**77.4**	**100**	**85.7**

Table 4.3: OSATS scores and GRS prediction results. Each cell contains two numbers in the form $\rho_{OSATS} \mid \rho_{GRS}$, where the first number is the value of ρ averaged over all OSATS and the latter is the value of ρ for GRS prediction. "*" means a $p-$value < 0.05 for the corresponding ρ.

	Suturing		Knot Tying		Needle Passing	
	LOSO	LOUO	LOSO	LOUO	LOSO	LOUO
SMT	0.25 | 0.46*	-0.08 | -0.28	0.41* | 0.39*	0.18 | 0.21	-0.12 | 0.09	0.07 | -0.60*
DCT	0.57* | 0.68*	0.10 | 0.08	0.59* | **0.76***	0.49 | 0.73*	0.22 | 0.26*	-0.16 | 0.09
DFT	0.45* | 0.49*	-0.28 | -0.29	0.31 | 0.32*	0.46* | 0.47*	0.44* | **0.53***	0.37 | 0.19
ApEn	0.31* | 0.49*	0.43 | 0.40*	0.26 | 0.14*	0.02 | 0.12	0.16 | 0.06	0.21 | -0.21
SMT+DCT	0.48* | 0.61*	0.01 | 0.01	**0.66*** | 0.71*	0.46 | **0.78***	0.14 | -0.16	-0.23 | -0.14
SMT+DFT	0.40* | 0.60*	-0.21 | -0.49*	0.36 | 0.39*	0.52* | 0.48*	0.39* | 0.54*	0.33 | 0.13
SMT+ApEn	0.28* | 0.35*	0.41 | **0.42***	0.18 | 0.36*	0.06 | 0.12	0.12 | -0.06	0.15 | -0.29
SMT+DCT+DFT	0.57* | 0.64*	0.16 | 0.10	0.58* | 0.70*	**0.56*** | 0.73*	0.36* | 0.38*	0.50* | 0.23
DCT+DFT	0.56* | 0.66*	0.13 | 0.14	0.53* | 0.68*	0.55* | 0.73*	0.41* | 0.47*	**0.53*** | **0.28**
DCT+DFT+ApEn	**0.59*** | **0.75***	0.43* | 0.37*	0.57* | 0.63*	0.48 | 0.60*	0.37 | 0.46*	0.23 | 0.25
SMT+DCT+DFT+ApEn	0.47* | 0.66*	**0.45*** | 0.37*	0.55* | 0.61*	0.49 | 0.62*	**0.45*** | 0.45*	-0.21 | -0.19

Table 4.4: Values of ρ averaged over all three tasks for the corresponding feature types in the form $\rho_{OSATS} \mid \rho_{GRS}$.

	LOSO	LOUO
SMT	0.18 | 0.31	0.05 | -0.22
DCT	0.46 | 0.57	0.14 | 0.24
DFT	0.40 | 0.45	0.19 | 0.12
ApEn	0.24 | 0.23	0.22 | 0.10
SMT+DCT	0.43 | 0.39	0.08 | 0.22
SMT+DFT	0.38 | 0.51	0.22 | 0.04
SMT+ApEn	0.20 | 0.22	0.21 | 0.08
SMT+DCT+DFT	0.50 | 0.57	**0.41** | 0.36
DCT+DFT	0.50 | 0.60	0.40 | 0.38
DCT+DFT+ApEn	**0.51** | **0.61**	0.38 | **0.41**
SMT+DCT+DFT+ApEn	0.49 | 0.58	0.24 | 0.27

Table 4.5: Root-mean-squared-error (RMSE) for each OSATS criteria using the top performing features from table 4.4. Each cell contains RMSE values for each task in the form Suturing | Knot tying | Needle passing.

	Respect for tissue	Suture handling	Time and motion	Flow of operation	Overall performance	Quality of final product
SMT+DCT+DFT	0.86 | 0.88 | 0.84	1.26 | 0.75 | 0.88	1.04 | 0.57 | 0.79	0.96 | 0.67 | 0.62	1.17 | 0.84 | 0.74	0.92 | 0.83 | 0.99
DCT+DFT	0.91 | 0.90 | 0.83	1.40 | 0.81 | 0.88	1.07 | 0.53 | 0.81	1.14 | 0.65 | 0.61	1.22 | 0.83 | 0.74	1.04 | 0.89 | 0.97
DCT+DFT+ApEn	0.88 | 0.93 | 0.90	1.02 | 0.96 | 1.21	0.90 | 0.55 | 0.82	0.86 | 0.74 | 0.71	1.02 | 0.92 | 1.11	0.83 | 1.12 | 1.17
SMT+DCT+DFT+ApEn	0.88 | 0.93 | 1.28	0.98 | 0.96 | 1.40	0.89 | 0.51 | 1.26	0.85 | 0.74 | 1.00	1.02 | 0.88 | 1.29	0.82 | 1.12 | 1.38

nificance using the p-value. The value of ρ can range from -1 to +1, where the more positive the value of ρ is, the more positively correlated the predicted and ground truth scores are (which we want in our case). For OSATS score prediction, we show the value of ρ averaged over all six criteria, whereas, the GRS ρ values are given as is. Feature combination results presented in Table 4.3 are evaluated using weighted feature fusion as described in methodology section. Overall, we can see that individual features and their combinations achieve good results for the LOSO setup. Specifically, DCT and DFT features perform better than others. On the other hand, we see a comparatively low performance overall across all feature combinations for LOUO setup with many negative values of ρ observed. This is because LOUO is a harder validation scheme due to less data for training phase. However, using the proposed feature combination significantly improves performance over individual features and results in a positive ρ for most feature combination cases. In general, frequency features seem to perform well when used individually or in combination with other features. We can also see an overall lower performance across all features for the needle-passing task. The reason for this could be that needle-passing is a relatively less repetitive task as compared to the other two. Since the features we use try to differentiate between different skill levels using data repeatability, they perform less well for needle-passing. Table 4.4 shows the average of ρ values over all three tasks (as given in Table 4.3) for each feature type. We observe that DCT+DFT+ApEn performs best on average for OSATS and GRS score prediction. We also evaluate the root-mean-squared-error (RMSE) values between the predicted and ground truth scores per OSATS criteria for the top performing features as shown in Table 4.5. We can see that the presented combination

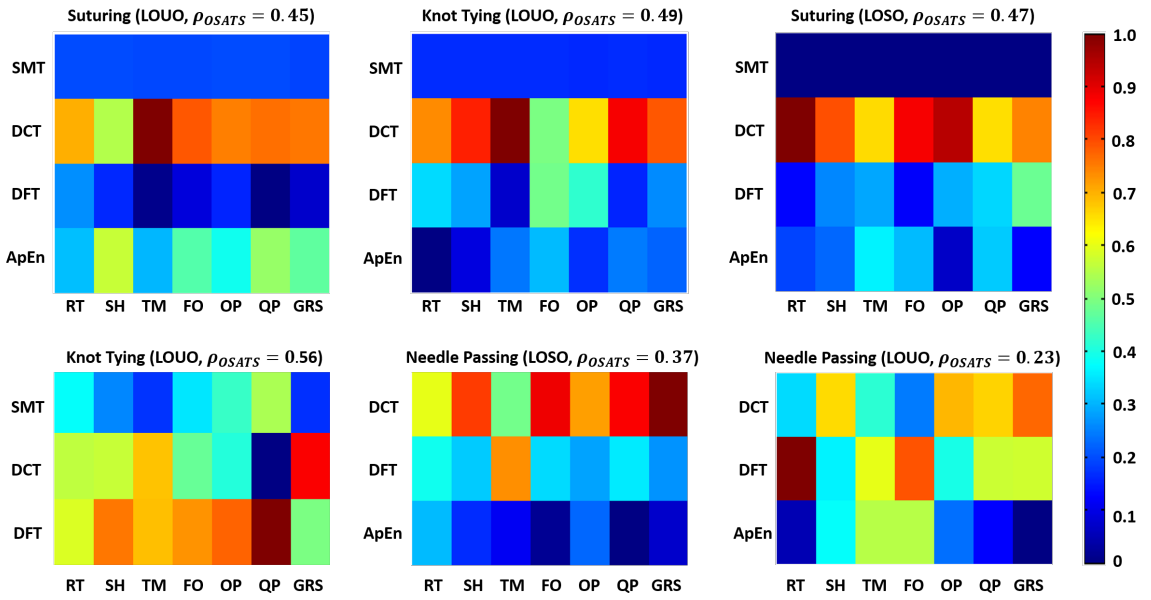

Figure 4.4: Heatmaps of weight assignments of different features. Each column shows the weight vector $w*$ (scaled from 0 to 1) for the corresponding OSATS criteria or GRS. For each heatmap, the features used in combination are shown next to each row and the corresponding task, validation scheme and average ρ (over OSATS) are also shown. (Please view this figure in color)

of features perform reasonably well for all the different criteria. This is interesting to see since one would expect that kinematics data alone may not be enough for some criteria like respect for tissue where visual information would be key in analyzing skill score. However, as confirmed by our results, robot kinematics data alone can potentially be enough for assessment of all OSATS criteria.

In order to analyze the role of different features in the proposed weighted late fusion for skill prediction, we generate heatmaps of the weight vectors learned and show a few of them in Figure 4.4. It can be seen that DCT features get assigned the highest weight in most of the cases. DFT and ApEn features generally have similar weight assignments whereas SMT always gets assigned a low weight. This shows that DCT features capture the most skill relevant information which is also evident from its high performance compared to other individual features in Table 4.3.

While the framework presented in this chapter show promising results for automated

surgical skills assessment for RMIS training, this work is limited by the amount of data that the analysis is performed on. JIGSAWS is the only publicly available dataset to date (to the best of our knowledge) for surgical skills assessment in RMIS training. Therefore, it is hard to claim that such methods would be generalizable. However, we believe that the idea of using predictability and fluency in surgical motions extracted through features like DCT, DFT and ApEn, should be able to differentiate skill reasonably well for other kinds of surgical data too.

4.4 Summary

In this chapter, we extended the framework presented in the previous chapter to RMIS training assessment and used holistic features like SMT, DCT, DFT and ApEn for skill assessment in RMIS training. The proposed framework out-performed all existing HMM based approaches. We also presented a detailed analysis of skill assessment on the JIGSAWS dataset and propose a weighted feature combination technique that further improved performance on score predictions. No video data was used making this method computationally feasible for real time feedback. This framework can easily be integrated in a robotic surgery platform (like the daVinci system) to generate automated OSATS based score reports in training.

CHAPTER 5

UNSUPERVISED SURGICAL ACTIVITY RECOGNITION

In the previous two chapter, we looked at methods that can be used to assess surgical skill in a basic training setup for open and robotic surgeries. The focus of this thesis will now shift towards assessment of robot-assisted clinical procedures which involves operating on real tissue - porcine or human. Assessment in such a setup becomes much harder since the environment is not controlled as before and every data sample would have large variations. Therefore, the frameworks presented previously cannot directly be applied for assessment in clinical procedures. The first problem that we need to solve for assessment in clinical setup is *'procedure segmentation'*. Procedure segmentation refers to finding the start and stop times of individual tasks within a procedure. This is an essential step for generating task wise assessment reports for surgeons. In this chapter, we will explore some unsupervised methods for procedure segmentation in robot-assisted surgeries (RAS).

5.1 Introduction

Over the course of entire procedures, surgeons perform certain tasks that are more critical than others. For example, during a prostatectomy, surgeons must finely coordinate their tools to carefully avoid damaging nerves during the dissection of the neurovascular bundles whereas mobilizing the colon and dropping the bladder do not involve similar risks. Despite these apparent differences across steps, most evaluations of surgical workflow or surgeon skill at population scales use simple, descriptive statistics (e.g. time) across whole procedures, thereby deemphasizing critical steps and potentially obscuring critical inefficiencies or skill deficiencies. If we could develop tools and algorithms to automatically recognize

Chapter reference: [52]

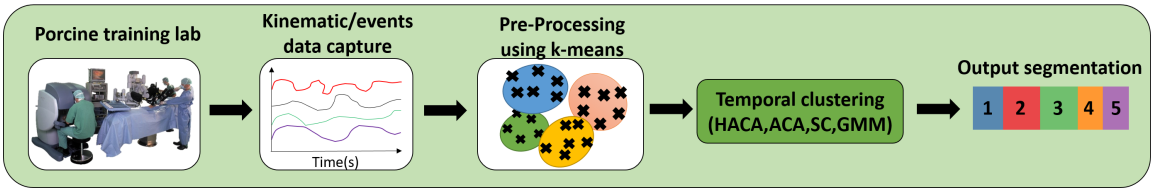

Figure 5.1: Flow diagram of the proposed model for unsupervised surgical phase segmentation.

clinically-relevant surgical tasks within procedures, we might be able to improve surgical workflow, skill assessment, surgeon training, and, ultimately, patient safety by providing task-specific performance measures.

Despite the recent successes of video-based methods, there remain compelling reasons why one would (a) want to use smaller data streams than video and (b) utilize offline methods without real-time capability. Small data streams enable feasible storage of data across many procedures, streaming of data over network connections without large bandwidth or disruption, and smaller compute resources for training the models. Using non-video data strongly parallels research directions in activity recognition where wearables with simple accelerometer signals might be used. Additionally, offline methods can utilize data from entire procedures for phase recognition and remain useful for post-operative feedback, review, and documentation by surgeons. For these reasons, we believe system data from robotic surgical systems offer a scalable, practical approach to surgical segmentation and skill estimation.

In this chapter, we will examine temporal clustering methods to perform offline surgical task recognition using only non-video data from RAS systems. In particular, we will apply models developed for human activity recognition [53, 54]. The models are evaluated on clinically relevant tasks performed on porcine models in a training environment.

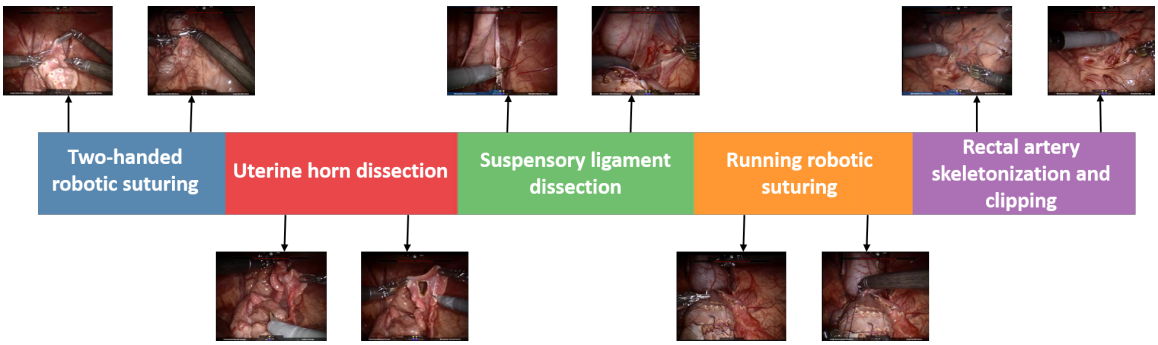

Figure 5.2: *'Pseudo-procedure'* with sample frames for each of the five surgical tasks in the dataset.

5.2 Methodology

In this section, we describe an approach for unsupervised segmentation of RAS procedures. Figure 5.1 shows a flow diagram of our method. We collect kinematic and events data from the da Vinci Si® surgical system (Intuitive Surgical, Inc., Sunnyvale, CA) while surgeons of varying expertise perform exercises on a porcine model (additional details on data set are given in Section 3). The events data stream is used directly, whereas, the kinematic time series is preprocessed before implementing different segmentation algorithms. We will employ Aligned Cluster Analysis (ACA) [53] and Hierarchical Aligned Cluster Analysis (HACA) [54] for our surgical procedure segmentation since both these algorithms have proven to work well for human activity segmentation. For comparison, we also implement two additional temporal clustering algorithms: Gaussian Mixture Models (GMM) and Spectral Clustering (SC). Descriptions of the clustering algorithms are given below.

5.2.1 Spectral Clustering

Spectral clustering (SC) is a graph based clustering algorithm which has been widely used for image segmentation in the computer vision community. It has also been used for time series segmentation in various biomedical applications [55]. For a given time series $T \in \Re^{d \times N}$, SC divides the temporal data depending on a similarity measure s_{ij} between pairs

of data frames t_i and t_j. The data is represented as a similarity graph $G = (V, E)$, where V is the vertex set and E is the edge set. Each vertex of the graph v_i is represented by a data frame t_i, and any two vertices are connected via a Gaussian similarity measure $s_{ij} = exp(-\frac{||t_i - t_j||^2}{2\sigma^2})$. Once the graph G is constructed, the problem of clustering becomes a graph partitioning task. Therefore, in order to cluster different surgical procedures in our dataset, we partition the graph constructed so that the edges between different groups have small weights and the edges within a group have large weights.

5.2.2 Gaussian Mixture Models

Gaussian mixture model (GMM) is a popular clustering algorithm and has been extensively used for various applications. The use of GMM for time series segmentation was originally proposed in [56]. We use a GMM to model our time series $T \in \Re^{d \times N}$, and segment the series whenever two consecutive frames belong to different Gaussian distributions. This is done since data frames from different surgical tasks, or activities in general, would potentially form distinct clusters which can be modeled using Gaussian distributions. We use the Expectation Maximization (EM) algorithm to estimate the parameters of each of the Gaussians in the GMM.

5.2.3 Aligned Cluster Analysis and Hierarchical Aligned Cluster Analysis

Given a time series $T \in \Re^{d \times N}$, Aligned Cluster Analysis (ACA) and Hierarchical Aligned Cluster Analysis (HACA) algorithms are formulated to decompose T into M different segments with each segment corresponding to one of the K clusters. Each segment Q_m consists of frames of data from position t_m till $t_{m+1} - 1$, where t_m and $t_{m+1} - 1$ represent the first and the last index of the mth segment. In order to control the temporal regularity, the length of each segment Q_m is constrained to the range $l_i \in [l_{min}, l_{max}]$. A binary indicator matrix $G \in \Re^{K \times M}$ is generated where $g_{k,m} = 1$ if the mth segment belongs to the kth cluster, otherwise $g_{k,m} = 0$. The objective function for the segmentation problem

is formulated as an extension to previous work on kernel k-means and is given by:

$$J_{ACA}(G, s) = \sum_{k=1}^{K} \sum_{m=1}^{M} g_{k,m} D_{\psi}^{2}(Q_m, z_k) \tag{5.1}$$

where, the distance function $D_{\psi}^{2}(Q_m, z_k) = ||\psi(T_{[t_i,t_i+1]}) - z_k||^2$, Q_m represents a segment, s is a vector containing the start and end of each segment and z_k is the geometric centroid of the k-th class. Just like kernel k-means, the distance between a segment and a class centroid is defined using a nonlinear mapping $\psi(.)$, given by

$$D_{\psi}^{2}(Q_m, z_k) = \tau_{mm} - \frac{2}{M_k} \sum_{j=1}^{M} g_{kj} \tau_{mj} + \frac{1}{M_k^2} \sum_{j_1, j_2=1}^{M} g_{kj_1} g_{kj_2} \tau_{j_1 j_2} \tag{5.2}$$

where, M_k denotes the number of segments belonging to class k. The dynamic kernel function τ is defined as $\tau_{ij} = \psi(Q_i)^T \psi(Q_j)$. In matrix form, the objective function for ACA can be written as

$$J_{ACA}(G, H) = tr((I_m - G^T (GG^T)^{-1} G) H (F \circ W) H^T) \tag{5.3}$$

where, W is the normalized correspondence matrix, H is the segment indicator matrix and F is the frame kernel matrix, as defined in [54]. For our analysis, frame kernel matrix is of particular interest since the preprocessing parameters depend on it. Given a time series $T \in \Re^{d \times N}$, the frame kernel matrix $F \in \Re^{N \times N}$ is given by

$$F = \phi(T)^T \phi(T) \tag{5.4}$$

Each element of the matrix f_{ij} represents the similarity between the corresponding frames, t_i and t_j, using a kernel function. We use a Gaussian kernel function for evaluating the frame kernel matrix giving $f_{ij} = exp(-\frac{||t_i - t_j||^2}{2\sigma^2})$. Once the energy function J_{ACA} is formulated, a dynamic programming based approach is used to solve for the optimal $G \in \Re^{K \times M}$ and $s \in \Re^{M+1}$ [54].

Table 5.1: Details of the five surgical tasks used in this study.

Task	Name	Mean Time (s)	Standard Deviation Time (s)
1	Two-handed robotic suturing	1329.2	733.9
2	Uterine horn dissection	2159.7	492.6
3	Suspensary ligament dissection	1999.3	1097.5
4	Running robotic suturing	617.6	126.7
5	Rectal artery skeletonization and clipping	1474.7	276.3

For Hierarchical aligned cluster analysis (HACA), the same steps as described above for ACA are performed in a hierarchy at different temporal scales reducing the computational complexity; HACA first searches in a smaller temporal scale and propagates the result to larger temporal scales. Temporal scales over here refers to the number of segments the time series is randomly segmented into initially; a larger scale would mean less number of segments. We use a two level HACA; the maximum segment length is restricted to $l_{max}^{(1)}$ and $l_{max}^{(2)}$ for the first and second levels in the hierarchy, respectively, where $l_{max}^{(1)} < l_{max}^{(2)}$. Please see [54] for a more detailed description of ACA and HACA.

5.3 Experimental Evaluation

5.3.1 Dataset

We collected data from nine RAS surgeons operating the da Vinci Si surgical system. Informed consent was obtained from all individual surgeons included in the study (Western IRB, Inc. Puyallup, WA). None of the surgeons had performed previous RAS procedures but they all had prior laparoscopic and/or open experience. Five of the surgeons specialized in general surgery, three specialized in urology, and one specialized in gynecology. Each of the surgeons performed multiple training tasks in a single sitting on a porcine model that focused on the technical skills used during dissection, retraction, and suturing. During each exercise, instrument kinematics, system events, and endoscope video were recorded and synchronized. System data was recorded at 50Hz whereas endoscope video was recorded at 25fps.

We selected five representative tasks for this study (see Table 5.1). The five tasks were treated as one 'pseudo-procedure' in our analysis as shown in Figure 5.2. The video data was used to generate ground truth segmentations and was not added as a source of features in our models. All tasks were performed in the pelvis of the porcine model and the setup joints (therefore, remote centers of motion) were unchanged for all tasks. The five tasks were perfomed on common anatomy within the pelvis thus ensuring that the segmentation algorithms are not simply using positions in the world reference frame to differentiate activities. Additional details about the instrument kinematic and system events data are given below.

Kinematic Data: The kinematic data captured from the da Vinci Si surgical system consisted of the endpoint pose and joint angles from the hand controllers on the surgeon side console (SSC) and the instruments and camera on the patient side cart (SI). The kinematic data stream from SSC consisted of a 56-dimensional time series whereas SI was a 156-dimensional time series. We used individual data streams along with their different combinations in order to find the data stream most useful for segmenting different surgical tasks.

Events: A subset of the available system events were used in this study. The events used included camera control, master clutch for each hand controller, instrument following state for three patient-side arms, energy activation, and surgeon head in/out of the console. All events were represented as binary on/off time series. In total, the events data was an 8-dimensional time series.

5.3.2 Parameter Estimation

The performance of each proposed clustering algorithm depends on various parameters at each step of the pipeline. We used a subset of 5 randomly selected 'pseudo-procedures' to estimate the different parameters empirically. The details are given below.

In the preprocessing step for kinematic data, we use k-means clustering per trial to

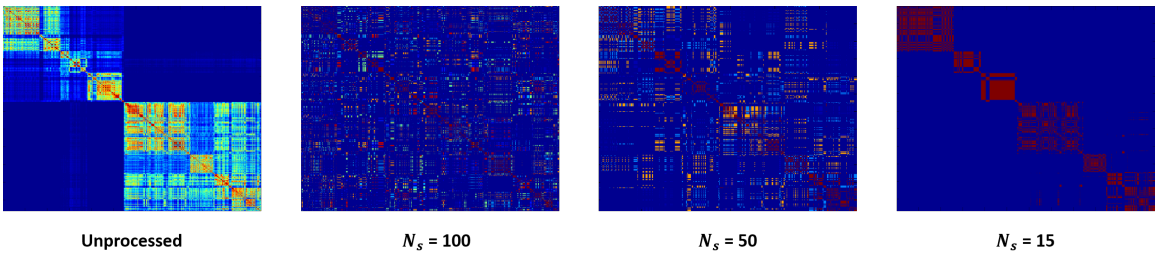

Figure 5.3: Sample frame kernel matrices for different number of symbols used in the preprocessing step. The left most image represents the frame kernel matrix when the time series is not reduced using k-means.

convert the high-dimensional time series data into symbols. The number of symbols, N_s, used in this step is important for the clustering performance since selecting too few symbols would fail in capturing enough information to differentiate the surgical tasks. The structure of the frame kernel matrix F, as described in Section 3, highly depends on the value of N_s. Ideally, in order to temporally segment different surgical tasks, we would want F to have a block structure along its diagonal. A block structure of K would mean a high variability in frames between different surgical tasks, and a low variability within each task. In [54], the authors selected the number of symbols (or clusters) based on characteristics of the synthetic or real data and made sure the chosen number of symbols was greater than the number of activities to be recognized. Here, we performed a coarse parameter search for the number symbols by running our clustering algorithms for a range of $N_s \in [10, 15, 20, 50, 100, 150, 200]$ and evaluated the clustering accuracies (using equation 6) for the selected subset of 'psuedo-procedures'. The value of N_s corresponding to the highest average clustering accuracy (over the subset of 'pseudo-procedures') was then selected. We found that having a smaller value of N_s gave better performance, with the highest average clustering accuracy being achieved with $N_s = 15$. Figure 5.3 shows example frame kernel matrices for the same time series data but with different value of N_s. One can see that using fewer symbols results in a more block-like structure in the frame kernel matrix. We used 15 symbols to represent our multi-dimensional time series before employing the temporal

clustering algorithms.

For ACA and HACA, the main parameter to fine tune is the maximum segment length l_{max}. ACA and HACA divides the time series into many small segments which are then assigned to different clusters. The lengths of these segments need to be selected in a way that maximizes segmentation performance. Keeping l_{max} too big would result in misclassifications at the boundaries between different tasks, whereas, a smaller l_{max} would not allow for the algorithm to capture the temporal structure of the data required for segmentation. In [54], the length constraints were again chosen based on characteristics of the datasets, siimilar to the number of clusters, N_s, without formal optimization. Therefore, we empirically selected the maximum segment lengths as $l_{max} = 30$ for ACA, and $l^1_{max} = 20$ and $l^2_{max} = 30$ for the two levels in HACA, respectively, based on the length of our tasks (see Table 5.1 and recording rate).

5.3.3 Evaluation Metric

In order to evaluate the clustering accuracy for each algorithm, we calculated the confusion matrix between the ground truth (G_{true}, H_{true}) and the segmentation output from the algorithm (G_{out}, H_{out}). The confusion matrix $C \in \Re^{K\times K}$ is given by:

$$C = G_{out} H_{out} H^T_{true} G^T_{true} \tag{5.5}$$

where, each element $c_{c_i c_j}$ represents the number of frames that are in cluster segment c_i and are shared by cluster segment c_j in ground truth. Once the confusion matrix is calculated, we use the Hungarian algorithm [57] to find the optimum cluster correspondence giving the clustering accuracy as:

$$accuracy = max \frac{tr(CP)}{tr(C1_{K\times K})} \tag{5.6}$$

where, $P \in \{0,1\}^{K\times K}$ is the permutation matrix and $1_{K\times K}$ represents a matrix of all 1

Table 5.2: Average performance with standard deviations for various feature types tested for the different clustering algorithms using the complete dataset of nine surgeons. The highest performance achieved across different features for each algorithm is shown in bold.

	SC	GMM	ACA	HACA
SSC	71.1 ± 16.4	50.6 ± 6.2	70.8 ± 17.1	79.0 ± 12.3
SI	80.6 ± 7.5	51.2 ± 7.0	82.7 ± 8.6	85.5 ± 8.3
SSC+SI	**84.1 ± 13.9**	**54.9 ± 6.5**	78.1 ± 15.8	82.3 ± 8.0
SSC+EVT	72.2 ± 14.9	53.6 ± 6.0	73.2 ± 13.9	73.9 ± 14.6
SI+EVT	81.9 ± 11.2	53.9 ± 6.2	**82.9 ± 11.0**	**88.0 ± 7.1**
SSC+SI+EVT	77.3 ± 17.6	52.3 ± 3.1	79.9 ± 14.0	84.1 ± 9.2

Table 5.3: Precision and recall values for different algorithms for each task using SI+EVT features.

	Precision				Recall			
	SC	GMM	ACA	HACA	SC	GMM	ACA	HACA
Task1	52.4	48.8	73.2	89.2	63.1	49.3	68.3	87.4
Task2	85.0	52.7	69.3	80.3	74.6	59.5	85.7	81.5
Task3	76.6	47.5	86.4	73.7	80.3	59.7	99.7	99.7
Task4	42.1	37.8	73.0	59.7	36.0	19.6	43.3	37.3
Task5	77.9	57.4	94.8	90.0	81.2	53.6	85.8	81.1

entries.

We employed the temporal clustering algorithms on individual data streams as well as their combinations. All possible combinations from these three data streams were evaluated to find the optimum features for our task. We computed the precision and recall for the top performing set of features based on the accuracy measures.

5.4 Results and Discussion

We evaluated the performance of the different unsupervised clustering algorithms on the surgical procedures. As described in Section 5.3.1, the dataset consisted of kinematic (pose and joint angles) and event data streams collected from the surgeon side console and the patient side cart. We implemented the clustering algorithms on individual data streams and combinations of different data streams in order to compare how various feature sets impacted algorithm performance. Since the convergence of clustering algorithms depends on

the initialization, we ran the algorithms for 5 different initializations and picked the solution with minimum energy (given by equation 3), which was the same methodology as [54]. Note that the solution that minimized the objective function also gave the highest clustering accuracy (evaluated using equation 6). Table 5.2 shows the mean accuracies achieved (over nine surgeons) for different algorithms and data streams used. Additionally, Table 5.3 shows the precision and recall values across tasks for the top performing data stream (SI+EVT). Task 4 consistently under performs compared to the other tasks across algorithm types. Furthermore, the mean F1 score for each algorithm was: SC (0.67), GMM (0.48), ACA (0.77), HACA (0.77). Based on these scores, ACA and HACA perform comparably but significantly outperfrom SC and GMM.

As a baseline comparison, we computed the segmentation accuracy when we simply scaled the normalized task lengths (relative to total procedure time) for each trial to estimate the transitions between tasks. The resulting accuracy is 0.60 (+/- 0.15) slightly better than GMM but worse than the remaining algorithms (see Table 5.2). This ensures the algorithms are not simply scaling tasks based on time. Although it serves a useful comparison, one can see from the example procedure bars (Figure 5.4) that the duration of tasks differed for different subjects.

From the results, we can see that SC, ACA and HACA perform fairly well while GMM performs poorly for all the feature types. As a whole, HACA out-performs all other methods for all but one feature type (SSC+SI). In general, using SSC kinematic data seems to perform less well than SI, which might be because SSC contains less information than SI (i.e., hand movements versus three instrument and camera movements). Adding EVT data to SSC and SI individually improves the segmentation accuracy for most of the algorithm types but deteriorates the performance when used with the combined kinematic data (SSC+SI). The highest accuracy achieved across all algorithms and features types was 88.0% using HACA with SI+EVT data. Results presented here are comparable to other surgical phase recognition methods in the literature [16, 32, 58].

Table 5.4: Average performance with standard deviations for each of the five tasks (T1 to T5). The feature set was SSC+SI+EVT.

	T1	T2	T3	T4	T5
ACA	66.4 ± 44.3	71.8 ± 33.9	84.8 ± 13.6	74.4 ± 45.9	94.6 ± 8.0
HACA	87.5 ± 35.4	81.2 ± 20.4	80.2 ± 20.4	61.7 ± 51.0	88.2 ± 26.4

Figure 5.4 shows example segmentation bars for four surgeons using the four different algorithms. The color scheme used for different surgical tasks in a procedure is the same as in Figure 5.2. For each surgeon, the five total rows corresponded to segmentation using ground truth, HACA, ACA, GMM and SC, respectively. One can see HACA outperforms the other methods, in general. Most misclassifications occur at the boundaries of tasks. Unlike other methods, GMM (and to some extent SC) made many misclassifications throughout each task. In some cases, we can achieve very accurate segmentation using HACA and ACA, as shown in the lowest block in Figure 5.4.

Finally, Table 5.4 shows the classification accuracy for each of the five tasks using ACA and HACA with the SSC+SI+EVT feature set. For ACA, the first tasks achieved the lowest accuracy whereas the fifth task achieved the highest accuracy. Conversely, for HACA the fourth task achieved the lowest accuracy whereas the fifth task achieved the highest accuracy. Across all tasks, HACA achieved a slightly more consistent classification accuracy. A one-way ANOVA showed that GMM, ACA, and HACA outperform SC across all feature types ($p < 0.01$). No significant differences existed between GMM, ACA, or HACA. A two-way ANOVA for algorithm type and features showed that both the algorithm and feature type affect accuracy ($p < 0.05$) but not their interaction. Additionally, a Friedmans test showed that algorithm type affects accuracy ($p < 0.001$).

Depending on the requirements for a particular end application, some misclassification error might be tolerable around task boundaries, especially at the task-level since the duration of tasks is on the order of minutes whereas the misclassification might be seconds. For example, compare the task boundaries between ground truth and HACA in the third surgeon in Figure 5.4; the relative amount of misclassified frames is much smaller than the

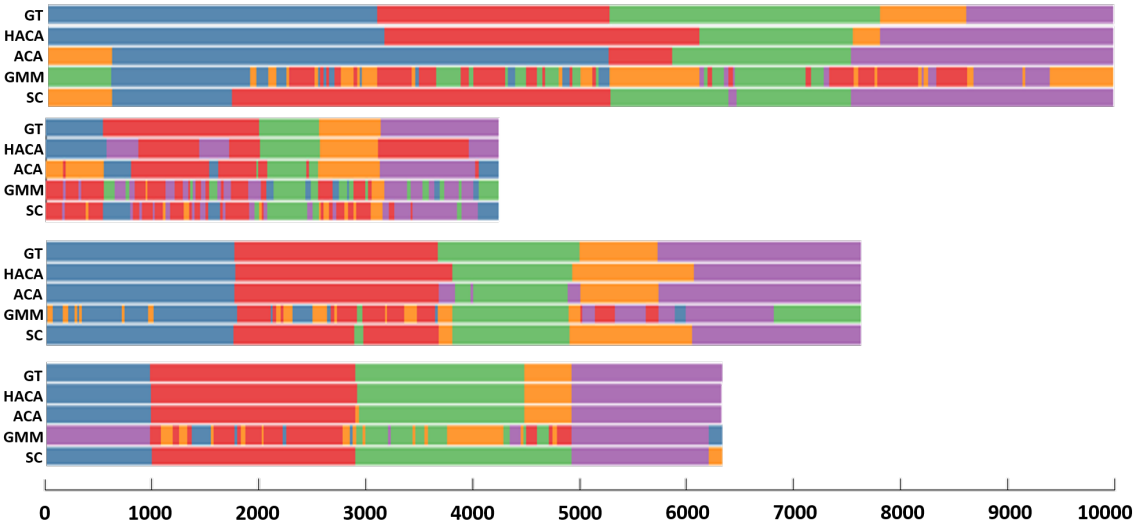

Figure 5.4: Segmentation results for four procedures. Each block contains five bars showing segmentation output using ground truth (GT), HACA, ACA, GMM and SC.

total width of each colored bar or task. In this way, the accuracies achieved by HACA (or ACA) could be sufficient for certain advanced analyses.

There are several limitations that exist with our analysis. Firstly, we used only five tasks to make up a procedure when most clinical procedures have more clinically discernible steps. Secondly, more formal methods could be used to optimize the parameters of the unsupervised clustering algorithms, such as a k-fold cross-validation. However, unlike supervised machine learning algorithms, the clustering algorithms used here are designed to be unsupervised and applied to situations where ground-truth labels might not be available. Another limitation is that features derived from video data were not used to meet the requirement of a lightweight data set. However, video-based features could be used to improve performance, especially when segmenting a larger number of tasks. The recent success of video-based segmentation methods also suggests it is a worthwhile endeavor [17, 32].

Despite these limitations, the results show that RAS system data can be used by temporal clustering algorithms to accurately segment surgically realistic tasks without directly

modeling low-level sub-tasks. We confirm that aligned clustering techniques (ACA and HACA) outperform conventional approaches like SC and GMM. Furthermore, we show that certain feature sets result in higher accuracies, and that a subset of all available features or data might be sufficient for certain applications.

5.5 Summary

In this chapter, we examined offline temporal clustering methods to recognize individual steps during clinically-relevant training procedures in RAS. The long term goal for this research is to provide increasingly more targeted assessment of surgical activities rather than whole procedure measures. This will enable advanced metrics to be used to benchmark and assess surgical workflow and surgeon proficiency. Our results suggest that offline clustering methods can be used to chunk whole surgical procedures into individual, clinically-relevant steps with competitive accuracies. Additionally, our approach is complementary to vision-based methods in that it uses system-based data streams present in RAS.

CHAPTER 6

SUPERVISED SURGICAL ACTIVITY RECOGNITION

In the previous chapter, we looked at some unsupervised methods for clustering surgical activities in robotic surgery training. While an unsupervised approach can have many advantages in terms of not needing annotations and large amounts of data, we still cannot recognize the task being performed via unsupervised approaches. In order to develop an intelligent system to generate automated score reports for robot-assisted surgeries, we don't just need to separate one task from the other in a procedure, but we also need to recognize what the individual tasks are.

In this chapter, we will present models to detect automatically the individual steps of robot-assisted radical prostatectomies (RARP). Our models break a RARP into its individual steps, which will enable us to provide tailored feedback to residents and fellows completing only a portion of a procedure and to produce task-specific efficiency metrics to correlate to certain outcomes. By examining real-world, clinical RARP data, this work builds foundational technology that can readily translate to have direct clinical impact.

6.1 Methodology

The rich amount of data that can be collected from the da Vinci (dV) surgical system (Intuitive Surgical, Inc., Sunnyvale, CA USA) enables multiple ways to explore recognition of the type of surgical tasks being performed during a procedure. Our development pipeline involves the following steps: (1) extraction of endoscopic video and dV surgical system data (kinematics and a subset of events), (2) design of deep learning based models for

Chapter reference: [59]

surgical task recognition, and (3) design of post-processing models to filter the initial procedure segmentation output to improve performance. We will now go into details of the different parts of our pipeline.

6.1.1 System data based models

The kind of hand and instrument movements surgeons make during procedures can be very indicative of what types of task they are performing. For example, a dissection task might involve static retraction and blunt dissection through in and out trajectories, whereas a suturing task might involve a lot of curved trajectories. Therefore, models that extract motion and event based features from dV surgical system data seem appropriate for task/activity recognition. We explore multiple Recurrent Neural Network (RNN) models using only system data given the recent success of RNNs to incorporate temporal sequences. Since there are multiple data streams coming from the dV surgical system, we employ two types of RNN architectures - *single stream* (SS) as shown in Figure 6.1, and *multi-stream* (MS) as shown in Figure 6.2. For SS, all data streams are concatenated together before feeding them into a RNN. Whereas, for MS, each data stream is fed into individual RNNs after which the outputs of each RNN are merged together using a fully-connected layer to produce predictions. For training both architecture types, we divide our procedure data into windows of length W. At test time, individual windows of the procedure are classified to produce the output segmentation.

6.1.2 Video based models

Apart from the kind of motions a surgeon makes, a lot of task representative information is available in the endoscopic video stream. Tasks which are in the beginning could generally look more *'yellow'* due to the fatty tissues, whereas tasks during the later part of the surgery could look much more *'red'* due to the presence of blood after dissection steps. Moreover,

the type and relative location of tools present in the image can also be very indicative of the step that the surgeon is performing. Therefore, we employ various image based convolutional neural networks (CNN) for recognizing surgical activity using video data. Within the CNNs domain, there are three types of CNN architectures that are popular and have been proved to work well for the purpose of recognition. The first type uses single images only with two-dimensional (2D) convolutions in the CNN architectures. Examples of such networks include VGG [60], ResNet [61] and InceptionV3 [62]. The second type of architecture uses a volume of images as input (e.g., 16 consecutive frames from the video) and employs three-dimensional (3D) convolutions instead of 2D (see Figure 6.4). C3D is an example of such model [63]. A potential advantage of 3D models is that they can learn spatio-temporal features from video data instead of just spatial features. However, this comes at the cost of requiring more data to train as well as longer overall training times. The third type of CNN architecture which has proved to work well recently by many works is a combination of CNN and RNN (see Figure 6.9). Multiple images a fed into individual CNN models to learn visual features. The extracted features are then concatenated together to be fed into an RNN in order to learn temporal features from the stream of images. In this type of a model, both spatial and temporal structure of the data is learned without having the difficulty of training models with 3D convolutions.

6.1.3 Video and system data based models

While single image and system data based models can potentially work great individually, a combination of both could help improve recognition scores significantly. Therefore, we also employ a combination of single image and multi-stream system based models (see Figure 6.5). A single image is fed into a CNN to extract visual features, while a preceding window of system data is fed into a multi-stream architecture of RNN as described above. The outputs of individual models are then merged together at the end using a fully connected layer.

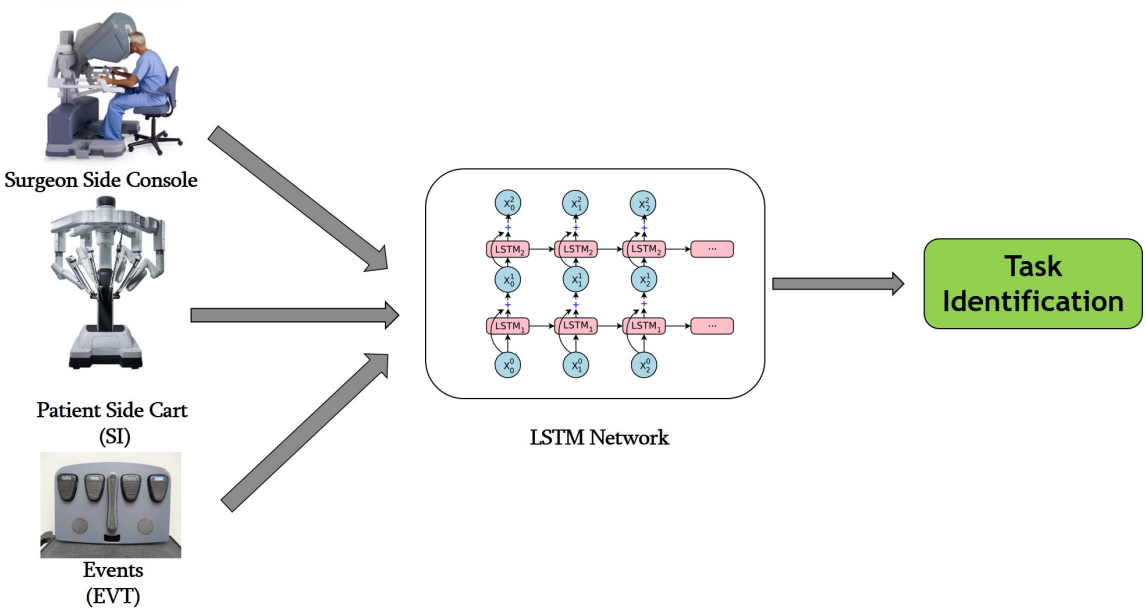

Figure 6.1: System data based single stream (SS) model for surgical task recognition

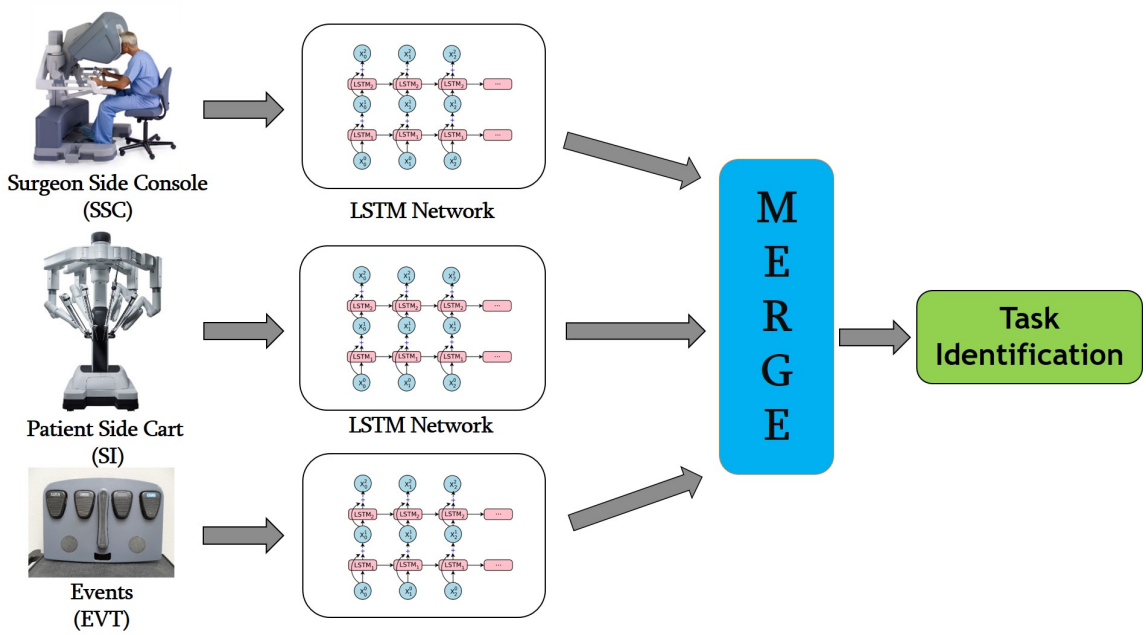

Figure 6.2: System data based multiple stream (MS) model for surgical task recognition

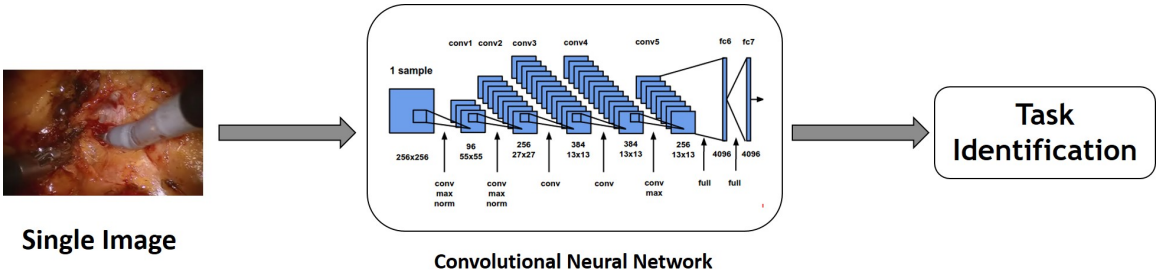

Figure 6.3: Single image based model for surgical task recognition

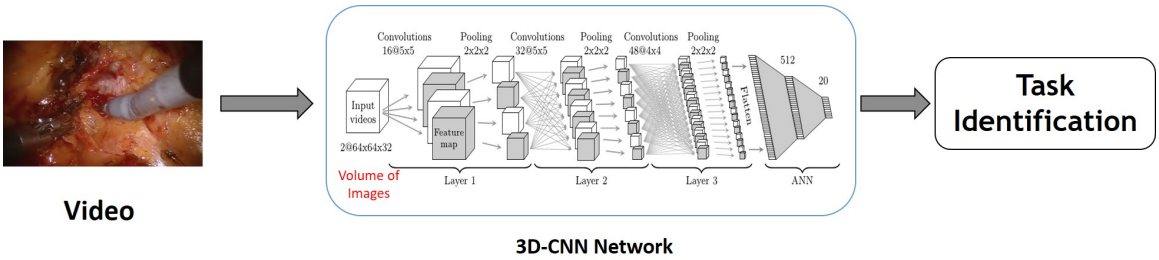

Figure 6.4: C3D network surgical task recognition

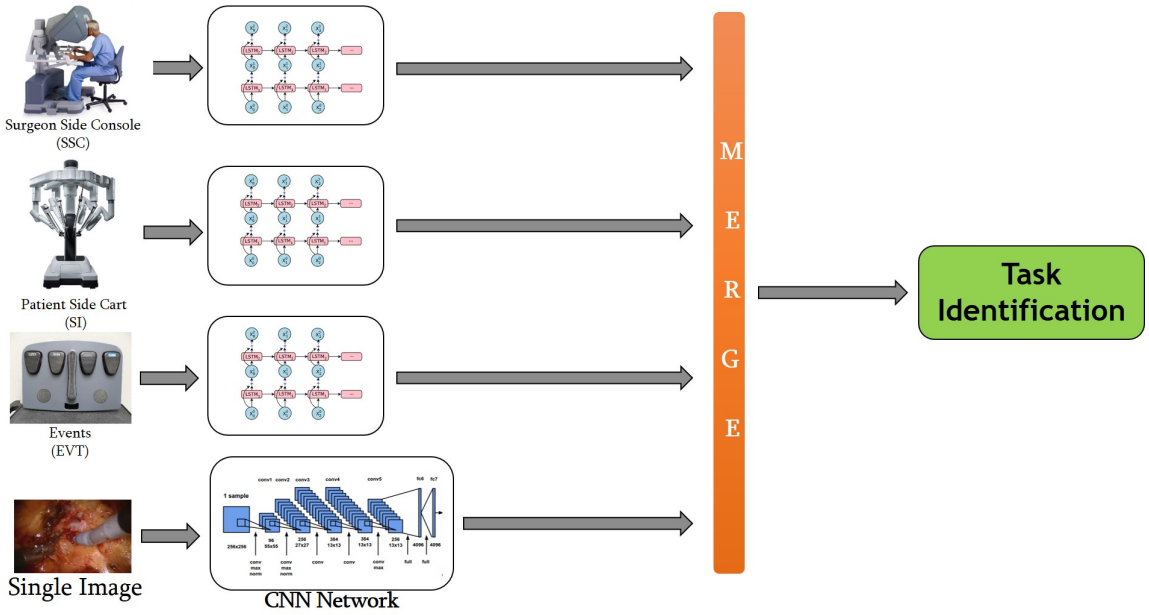

Figure 6.5: System data and single image model for surgical task recognition

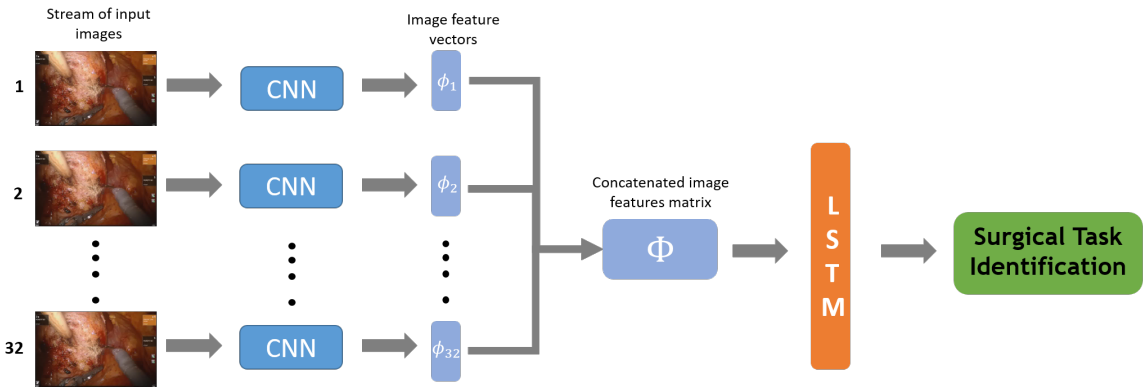

Figure 6.6: Multiple image CNN+LSTM model for surgical task recognition

6.1.4 Post-Processing

Since there are parts of various tasks that are very similar visually and in terms of motions the surgeon is making, the predicted procedure segmentation can have *'spikes'* of mis-classifications. However, it can be assumed that the predicted labels would be consistent within a small window. Therefore, in order to remove such noise from the output, we employ a simple running window median filter of length F as a post-processing step. For corner cases, we append the start and end of the predicted sequence with the median of first and last window of length F, respectively, in order to avoid mis-classifications of the corner cases by appending zeros.

6.2 Experimental Evaluation

6.2.1 Dataset

We collected a dataset consisting of 100 robot-assisted radical prostatectomies (RP) completed at an academic hospital. The majority of procedures were completed by a combination of residents, fellows, and attending surgeons. Each RP was broken into approximately 12 standardized tasks. The order of these 12 tasks varied slightly based on surgeon preference. The steps of each RP were annotated by one resident. A total of 1195 individual

Table 6.1: Dataset: the 12 steps of robot-assisted radical prostatectomy and general statistics.

Task no	Task Name	Mean time (sec)	Number of samples
T1	mobilize colon / drop bladder	1063.2	100
T2	Endopelvic fascia	764.2	98
T3	Anterior bladder neck dissection	164.9	98
T4	Posterior bladder neck dissection	617.5	100
T5	Seminal vesicles	686.8	100
T6	Posterior plane / Denonvilliers	171.2	99
T7	Predicles / nerve sparing	510.6	100
T8	Apical dissection	401.1	100
T9	Posterior anastomosis	403.1	100
T10	Anterior anastomosis	539.7	100
T11	Lymph node dissection Left	999.6	100
T12	Lymph node dissection Right	1103.6	100

tasks were used. Table 6.1 shows general statistics of our dataset.

Each RP recording included one channel of endoscopic video, dV surgical system kinematic data (e.g., joint angles, endpoint pose) collected at 50Hz, and dV surgical system event data (e.g., camera movement start/stop, energy application on/off).

The dV surgical system kinematic data originated from the surgeon console (SSC) and the patient side cart (SI). For both the SSC and SI, the joint angles for each manipulandum and the endpoint pose of the hand controller or instrument were used. In total, there were 80 feature dimensions for SSC and 90 feature dimensions for SI. The dV surgical system event data (EVT) consisted of many events relating to surgeon interactions with the dV surgical system originating at the SSC or SI. In total, there were 87 feature dimensions for EVT.

6.2.2 Data preparation

Several pre-processing steps were implemented for system and video data before they could be fed into the models. The endoscopic video was downsampled to 1 frame per second (fps) resulting in 1.4 million images in total. Image resizing and rescaling was model specific.

All kinematic data was downsampled by a factor of 10 (from 50Hz to 5Hz). Different window lengths (in terms of the number of samples) W (50, 100, 200 and 300) were tried for training the models and $W = 200$ performed the best. We use zero overlap when selecting windows for both training and testing. Mean normalization was also applied to all feature dimensions for the kinematic data. All events from the dV surgical system data that occurred within each window W were used as input for to our models. The events were represented as a unique integers with corresponding timestamps.

6.2.3 Model training and parameter selection

For RNN based models, we implement both SS and MS architectures for all possible combinations of the three data streams (SSC, SI, and EVT). Estimation of model hyperparameters was done via a grid search on the number of hidden layers (1 or 2), type of RNN unit (Vanilla, GRU or LSTM), number of hidden units per layer (8, 16, 32, 64, 128, 256, 512 or 1024) and what dropout ratio to use (0, 0.2 or 0.5). For each parameter set, we also compare forward and bi-directional RNN.

In all CNN based models, we used two approaches - training the networks from randomly initialized weights and fine-tuning the networks from pre-trained weights. For all models, we found that fine-tuning was much faster and achieved better accuracies. For single image based models, we used ImageNet [64] pretrained weights while for C3D we used Sports-1M [65] pretrained weights. We found that fine-tuning several of the last convolutional layers led to the best performances across models.

For both RNN- and CNN-based models, the dataset was split to include 70 procedures for training, 10 procedures for validation, and 20 procedures for test.

For the post-processing step, we evaluated performances of all models for values of F (median filter length) ranging from 3 to 2001, and choose a window length that led to maximum increase in model performance across different methods. The final value of F

was set to 301. All parameters were selected based on the validation accuracy.

6.2.4 Evaluation Metrics

For a given series of ground truth labels $G \in \Re^N$ and predictions $P \in \Re^N$, where N is the length of a procedure, we evaluate multiple metrics for comparing the performance of various models. These include average precision (AP), average recall (AR), F-score and Jaccard index. Precision is evaluated using $P = \frac{tp}{tp+fp}$, recall using $R = \frac{tp}{tp+fn}$ and Jaccard index using $J = \frac{tp}{tp+fp+fn}$, where tp, fp and fn represent the true positives, false positives and false negatives, respectively.

6.3 Results and Discussion

In order to compare the performance of different models, we evaluate all types of evaluation metrics presented above for all models. The results for all models without post-processing are shown in Table 6.2, while Table 6.3 shows results after post-processing. We can see that the multiple images CNN+LSTM models work best as compared to all others. In general, we observed that the image-based CNN models (except for C3D) performed better than the RNN models. Within LSTM models, MS architecture performed slightly better than SS with the SSC+EVT combination achieving the best performance. For nearly all models, post-processing significantly improved task recognition performance.

Figures 6.7, 6.8 and 6.9 show confusions matrices when using best system data based model, best single image model and best multiple image model, respectively, after post-processing. It can be seen clearly why system data based models do not perform too well - some of the tasks are almost always misclassified. This could be due to the fact that some tasks have very similar motions made by the surgeons. However, we do note that some tasks are reasonably well classified.

Single image and multiple image based models perform much better as a whole and on individual tasks as evident from a more 'diagonal nature' of the confusion matrices. There

Table 6.2: Surgical procedure segmentation results using different models. Each cell shows the average evaluations metric values across all procedures and tasks in the test set. For LSTM models, the modalities used are given in parentheses while the architecture type used is given in square brackets. Best performing model is shown in bold.

Model Type	Precision	Recall	Fscore	Jaccard Index
KIN_LSTM_FWD(SSC)	0.645	0.56	0.582	0.637
KIN_LSTM_BI(SSC)	0.642	0.6	0.607	0.652
KIN_LSTM_FWD(SI)	0.18	0.191	0.177	0.258
KIN_LSTM_BI(SI)	0.215	0.207	0.204	0.265
KIN_LSTM_FWD(EVT)	0.042	0.083	0.022	0.146
KIN_LSTM_BI(SI)	0.087	0.127	0.069	0.189
KIN_LSTM_FWD(SSC+SI)[MS]	0.591	0.555	0.56	0.613
KIN_LSTM_BI(SSC+SI)[MS]	0.585	0.565	0.559	0.629
KIN_LSTM_FWD(SSC+SI)[SS]	0.53	0.476	0.486	0.543
KIN_LSTM_BI(SSC+SI)[SS]	0.559	0.526	0.533	0.582
KIN_LSTM_FWD(SSC+EVT)[MS]	0.613	0.513	0.519	0.603
KIN_LSTM_BI(SSC+EVT)[MS]	0.625	0.572	0.586	0.633
KIN_LSTM_FWD(SSC+EVT)[SS]	0.589	0.508	0.527	0.593
KIN_LSTM_BI(SSC+EVT)[SS]	0.625	0.567	0.571	0.625
KIN_LSTM_FWD(SI+EVT)[MS]	0.212	0.204	0.165	0.262
KIN_LSTM_BI(SI+EVT)[MS]	0.223	0.212	0.202	0.249
KIN_LSTM_FWD(SI+EVT)[SS]	0.243	0.216	0.208	0.27
KIN_LSTM_BI(SI+EVT)[SS]	0.26	0.243	0.242	0.291
KIN_LSTM_FWD(SSC+SI+EVT)[MS]	0.569	0.527	0.518	0.583
KIN_LSTM_BI(SSC+SI+EVT)[MS]	0.437	0.446	0.405	0.552
KIN_LSTM_FWD(SSC+SI+EVT)[SS]	0.519	0.481	0.485	0.552
KIN_LSTM_BI(SSC+SI+EVT)[SS]	0.544	0.518	0.524	0.575
SINGLE_IMAGE(VGG-16)	0.633	0.535	0.541	0.614
SINGLE_IMAGE(VGG-19)	0.549	0.481	0.473	0.529
SINGLE_IMAGE(RESNET-50)	0.621	0.582	0.573	0.622
SINGLE_IMAGE(INCEPTION-V3)	0.662	0.642	0.632	0.666
C3D	0.569	0.535	0.538	0.623
MULTIPLE_IMAGES (INCEPTION-V3)	0.764	0.755	0.774	0.790
MULTIPLE_IMAGES (VGG-19)	**0.801**	**0.798**	**0.803**	**0.811**

Table 6.3: Surgical procedure segmentation results using different models after median filtering post-processing Each cell shows the average evaluations metric values across all procedures and tasks in the test set. For LSTM models, the modalities used are given in parentheses while the architecture type used is given in square brackets. Best performing model is shown in bold.

Model Type	Precision	Recall	Fscore	Jaccard Index
KIN_LSTM_FWD(SSC)	0.674	0.581	0.605	0.665
KIN_LSTM_BI(SSC)	0.664	0.625	0.63	0.681
KIN_LSTM_FWD(SI)	0.181	0.189	0.175	0.259
KIN_LSTM_BI(SI)	0.217	0.21	0.207	0.271
KIN_LSTM_FWD(EVT)	0.012	0.083	0.021	0.147
KIN_LSTM_BI(SI)	0.146	0.127	0.064	0.195
KIN_LSTM_FWD(SSC+SI)[MS]	0.609	0.573	0.578	0.638
KIN_LSTM_BI(SSC+SI)[MS]	0.595	0.572	0.566	0.645
KIN_LSTM_FWD(SSC+SI)[SS]	0.564	0.502	0.512	0.573
KIN_LSTM_BI(SSC+SI)[SS]	0.578	0.551	0.554	0.606
KIN_LSTM_FWD(SSC+EVT)[MS]	0.648	0.539	0.545	0.632
KIN_LSTM_BI(SSC+EVT)[MS]	0.648	0.593	0.609	0.662
KIN_LSTM_FWD(SSC+EVT)[SS]	0.623	0.528	0.549	0.614
KIN_LSTM_BI(SSC+EVT)[SS]	0.641	0.593	0.59	0.651
KIN_LSTM_FWD(SI+EVT)[MS]	0.213	0.202	0.164	0.261
KIN_LSTM_BI(SI+EVT)[MS]	0.221	0.209	0.198	0.25
KIN_LSTM_FWD(SI+EVT)[SS]	0.274	0.223	0.217	0.277
KIN_LSTM_BI(SI+EVT)[SS]	0.268	0.246	0.244	0.296
KIN_LSTM_FWD(SSC+SI+EVT)[MS]	0.595	0.547	0.537	0.606
KIN_LSTM_BI(SSC+SI+EVT)[MS]	0.458	0.471	0.431	0.582
KIN_LSTM_FWD(SSC+SI+EVT)[SS]	0.555	0.503	0.508	0.581
KIN_LSTM_BI(SSC+SI+EVT)[SS]	0.579	0.546	0.553	0.603
SINGLE_IMAGE(VGG-16)	0.747	0.621	0.627	0.715
SINGLE_IMAGE(VGG-19)	0.695	0.573	0.568	0.634
SINGLE_IMAGE(RESNET-50)	0.713	0.673	0.663	0.728
SINGLE_IMAGE(INCEPTION-V3)	0.782	0.759	0.749	0.786
C3D	0.352	0.367	0.329	0.418
MULTIPLE_IMAGES (INCEPTION-V3)	0.798	0.815	0.820	0.830
MULTIPLE_IMAGES (VGG-19)	**0.841**	**0.835**	**0.831**	**0.850**

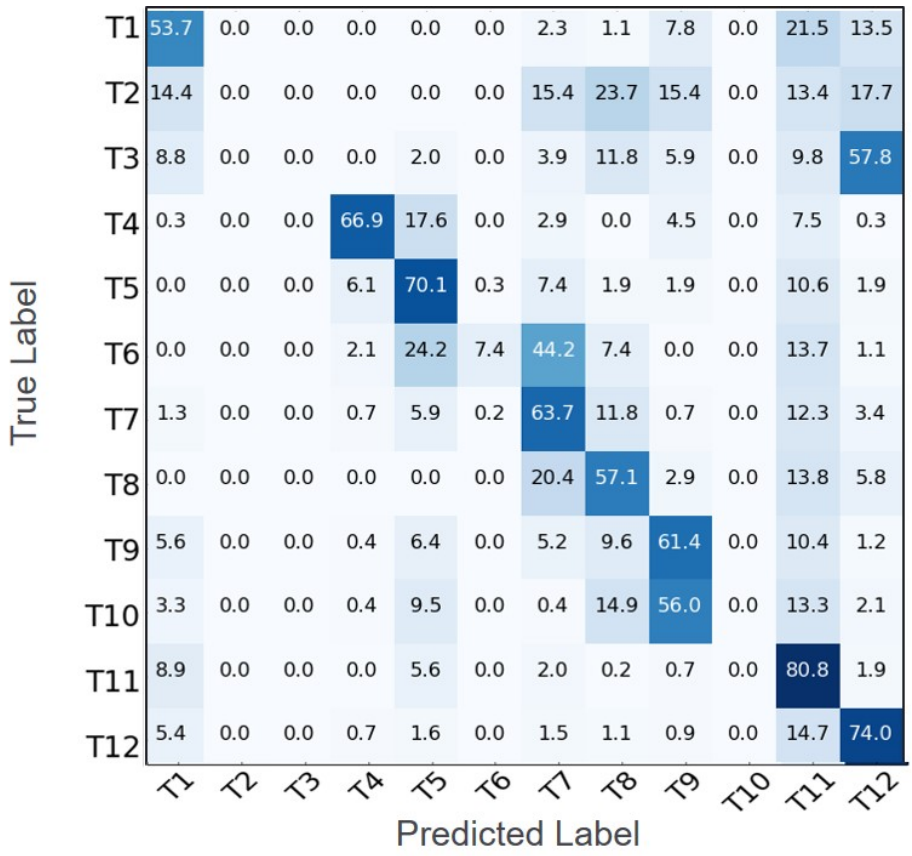

Figure 6.7: Confusion matrix for best system data based model.

are some interesting insights that we get from the single image model confusion matrix in Figure 6.8. The model performed well for almost all the tasks individually except for task 9. However, we can see that most of the task 9 samples were classified as task 10. Tasks 9 and 10 are very related - they are two parts of one overall task (posterior and anterior anastomosis). Furthermore, the images from these two tasks were quite similar given they show anatomy during reconstruction after extensive dissection and energy application. Hence, one would expect that the model could be confused on these two tasks. This is also the case for tasks 3 and 4 - anterior and posterior bladder neck dissection, respectively. When using the multiple image models, we see all individual tasks classification accuracies go up. However, the problem of confusion between similar looking tasks as that in the single image based models still remained.

Figures 6.10, 6.11 and 6.12 show visualizations of the segmentation results as color-

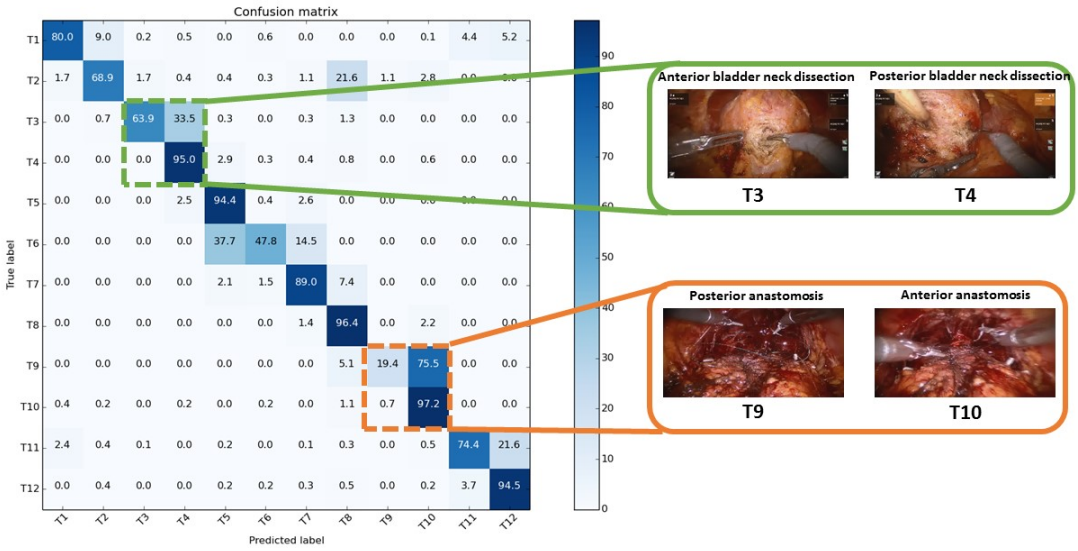

Figure 6.8: Confusion matrix of results using single image based model (Inception-V3) with post-processing. Sample images of tasks between which there is a lot of *'confusion'* are also shown.

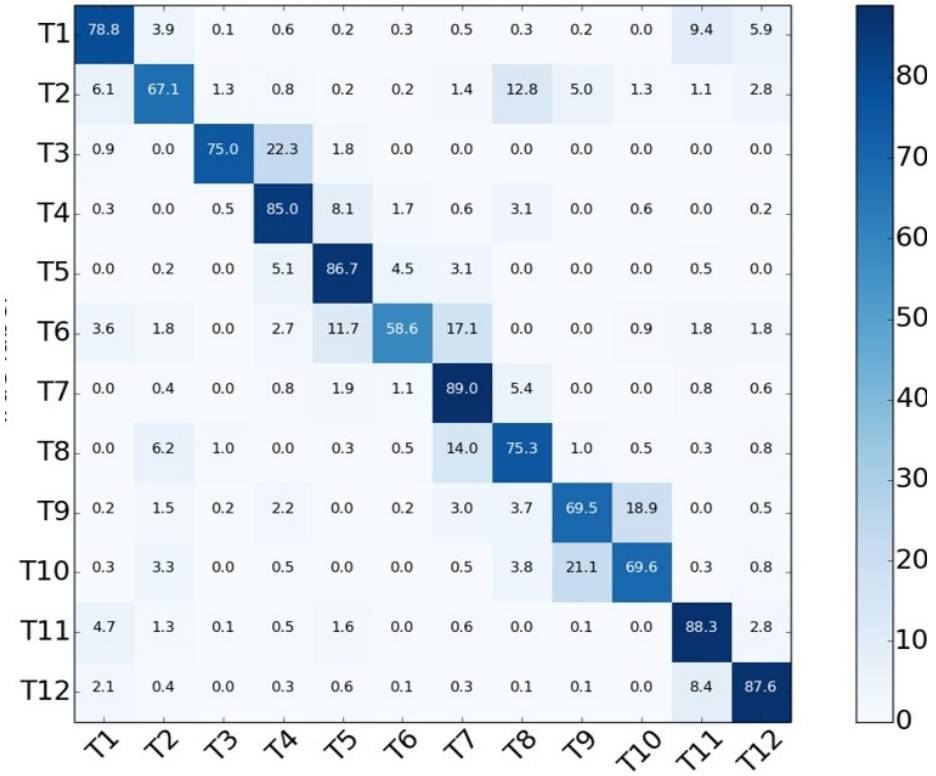

Figure 6.9: Confusion matrix for multiple images CNN+LSTM model

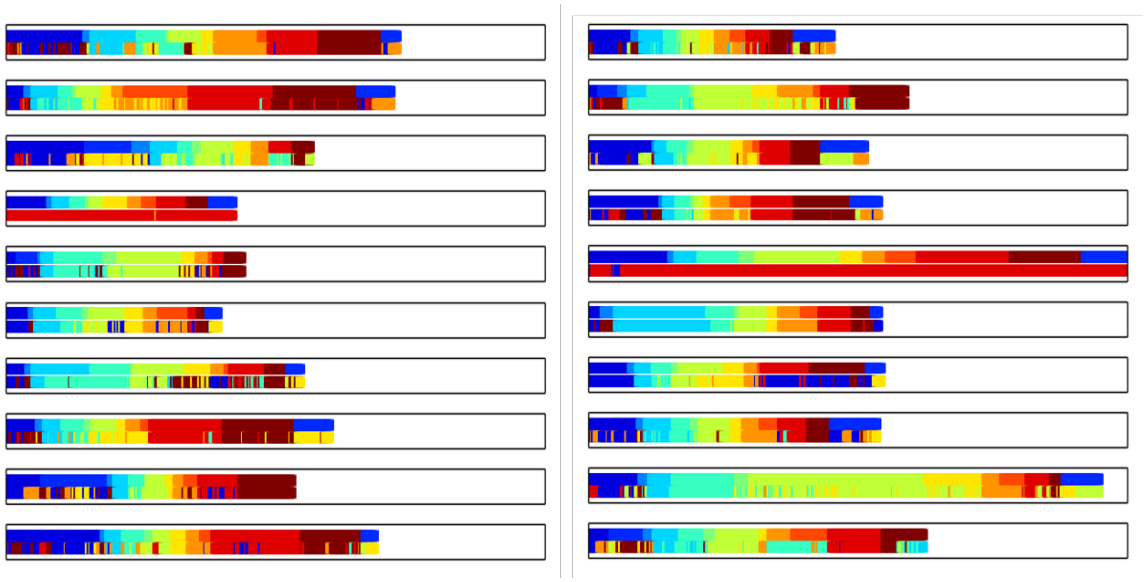

Figure 6.10: Segmentation bar plots for using best system data based model. Each box shows bar plots for one complete procedure with top half showing the ground truth and lower half showing the predictions. Each task is represented by a different color.

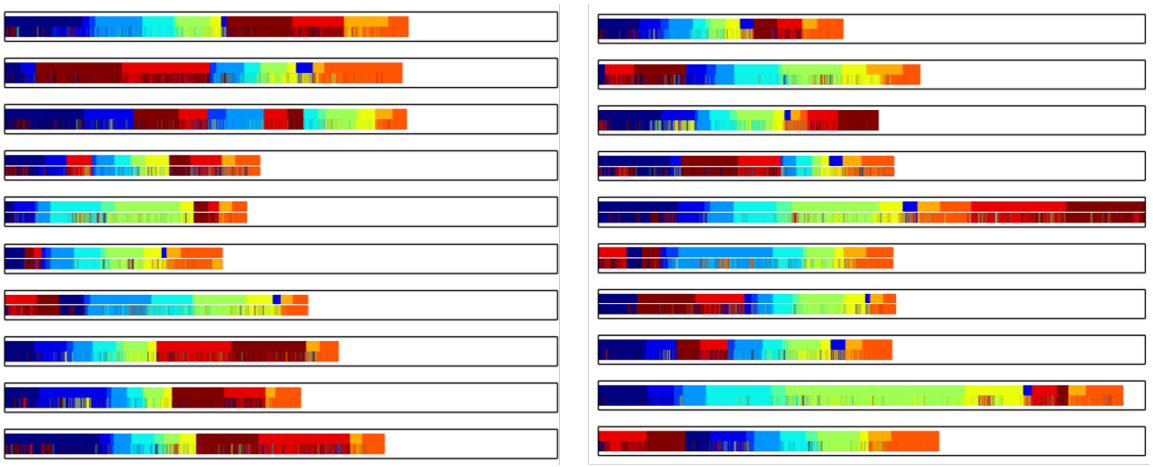

Figure 6.11: Segmentation bar plots for using single image model (Inception-V3). Each box shows bar plots for one complete procedure with top half showing the ground truth and lower half showing the predicitons. Each task is represented by a different color.

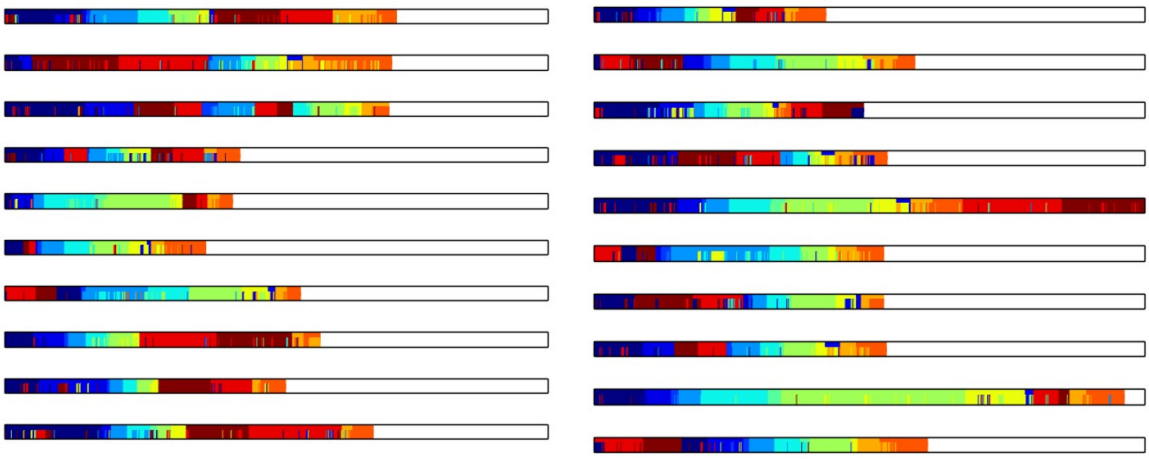

Figure 6.12: Segmentation bar plots for using multiple image CNN+LSTM model. Each box shows bar plots for one complete procedure with top half showing the ground truth and lower half showing the predicitons. Each task is represented by a different color.

coded bars when using best system data based model, best single image model and best multiple image model, respectively. In each figure, the individual boxes show segmentation bars for one complete procedure from the test set. Within each box, the top half shows the ground truth while the bottom half shows the predictions - each task is represented by a different color. As expected, system data based models segmentation outputs do not look good - some procedures are completely misclassified as well. However, this could be due to data acquisition errors as well since we do see tasks being somewhat classified correctly in almost all other cases.

The segmentation plots for image based models look much better. However, looking specifically to single image model outpout (Figure 6.11), we see undesired spikes in the predicted surgical phase. This can be explained by the fact that the model has no temporal information and classifies only using a single image which can lead to mis-classifications since different tasks can look similar at certain points in time. Multiple image model output (Figure 6.12) do look better and have lesser number of spikes, but its not completely eradicated. However, using the proposed median filter for post-processing significantly removes such noise and produces a more consistent output. Figure 6.13 shows how me-

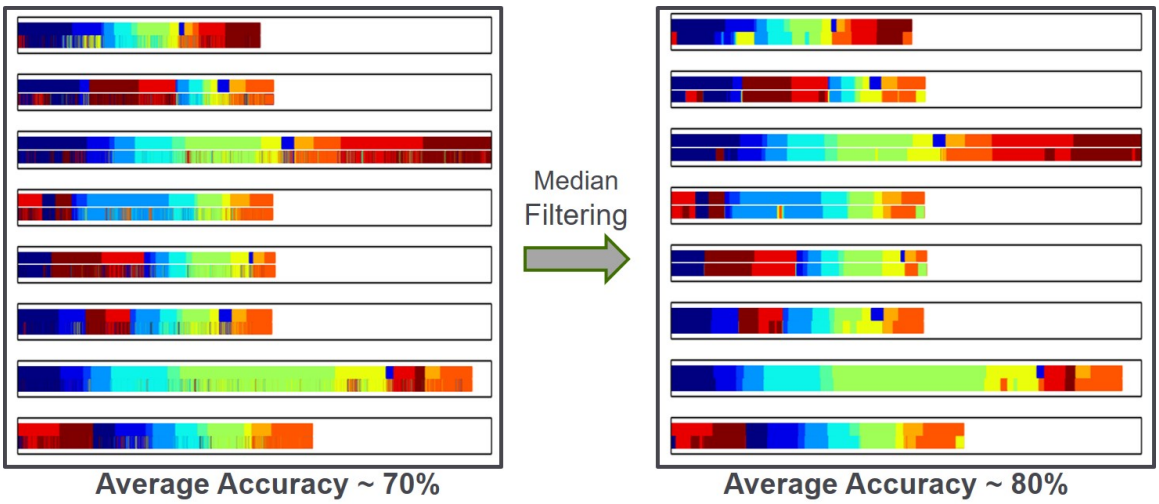

Figure 6.13: Improvements achieved on segmentation outputs using median filtering. The left part shows segmentation bars for a few cases without post-processing while the right one show after post-processing.

dian filtering improves model performance. As can be seen, the segmentation outputs after post-processing look much more clean and continuous without unwanted spikes. While the proposed median filtering resulted in a tremendous improvement in performance, there is a lot more that can be done in post-processing that can further improve results. For example, having temporal models like HMM on top of median filtering can potentially remove even more misclassification spikes that we get in the middle.

Despite not having temporal motion information, single image-based models recognize surgical tasks quite well. One reason for this result could be due to the significantly large dataset available for single-image based models. Given the presented RNN and C3D models use a window from the overall task as input, the amount of training data available for such models reduces by a factor of the length of window segment. Also, in general, RNN and 3D-convolitional models are harder to train. Though using CNN+LSTM model does have the same issue of less data, having 2D convoltional model for visual feature extraction seems to have made the training and results much better.

6.4 Summary

In this chapter, we presented various deep learning models to recognize the steps of robot-assisted radical prostatectomy (RARP). We used a clinically-relevant dataset of 100 RARPs from one academic center which enables translation of our models to directly impact real-world surgeon training and medical research. In general, we showed that image-based models outperformed models using only kinematic and events data. Having reasonably high accuracies on surgical activity recognition gives a good foundation to perform skill analysis on robot-assisted surgeries.

CHAPTER 7
AUTOMATED PERFORMANCE REPORT GENERATION FOR ROBOT-ASSISTED PROCEDURES

7.1 Introduction

A primary goal of surgeons is to minimize adverse outcomes while successfully treating patients. Although many factors can influence outcomes, the technical skills of surgeons is one area shown to correlate. Therefore, methods to evaluate technical skills are critical.

The most common approach for surgeon technical skill evaluation is expert feedback either intra-operatively in real-time or post-operatively through video review. However, an attending may not always be able to provide feedback in person, and post-operative video review can be time consuming and subjective. It is apparent this approach is not scalable, especially given the limited free time in expert surgeon schedules. Recently, crowd-sourced video evaluations has shown promise but this approach still faces scalability and accuracy concerns. Automated, objective, and less time-consuming methods to evaluate surgeon technical skills are needed.

Recently, objective, efficiency metrics derived from robotic-assisted surgical platforms have been defined for particular tasks within clinical procedures [66]. They have been shown to differentiate expertise and to preliminarily correlate to patient outcomes [67]. These basic efficiency metrics closely resemble those from virtual reality simulators and stand to offer a scalable method for objective surgeon feedback. However, there remain two primary challenges for these objective metrics to impact surgery. The first challenge is to define which metrics matter most for individual surgical tasks. Efficiency metrics must be defined to control for a large amount of variability including anatomical variations and surgical approach or judgment. Such variation is not present in virtual reality tasks

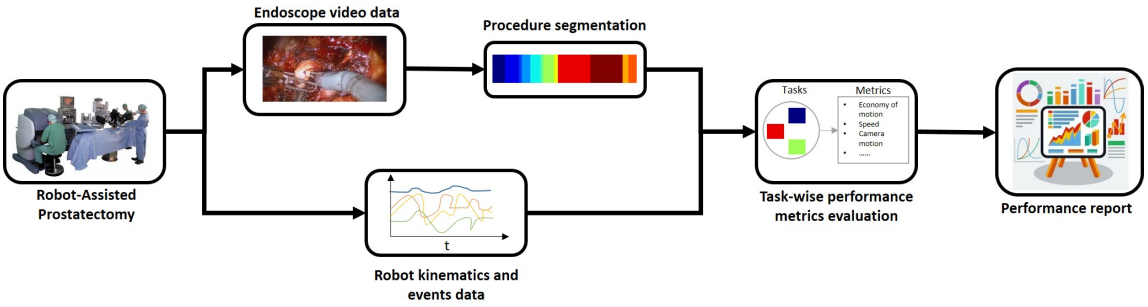

Figure 7.1: Flow diagram for automated performance report generation

where these metrics are common since only a handful of standardized exercises is available. In pursuit of this first challenge, several clinical research teams are actively working to discover and validate efficiency metrics in clinical scenarios. The second challenge is to automatically recognize the boundaries of surgical tasks. The begin and end times of tasks or sub-tasks must be automatically identified from within the entire procedure because manual identification through post-operative video review is overly time consuming and not scalable. Machine learning algorithms have been used with promising initial results in laparoscopic , retinal , and robotic-assisted surgeries .

A critical shortcoming of the prior work in automatic, surgical activity recognition is that the models have been evaluated solely by their frame-by-frame accuracy but not their impact on task-based, efficiency metrics. Since advanced intra-operative or post-operative feedback through efficiency metrics is the overall goal, the requirements and specifications of machine learning models should be at least in part defined by their ability to result in accurate metrics. Certainly if the models perfectly predict every frame of a surgical procedure, they can also be used to perfectly compute efficiency metrics.

In this chapter, we will use the procedure segmentation outputs produced in the previous chapter to generate automated performance reports in RARP. We will also explore new methods to quantify the effects of surgical activity recognition models on technical, efficiency metrics during clinical tasks.

7.2 Methodology

The process to generate automated performance reports for robotic-assisted surgery is shown in 7.1. First, procedure segmentation models (presented in previous chapter) chunk a surgery into individual tasks. Next, the relevant efficiency metrics are computed using robotic system data. Finally, a report can be composed incorporating metrics for individual surgical tasks. Each part of the pipeline is discussed below.

Procedure Segmentation: We use the multiple image model with VGG-19 (best performing for procedure segmentation) with post-processing from chapter 6 to recognize surgical activity form the raw video feed during an RARP. Despite post-processing, misclassifications can still exist causing cases where the predictions have disconnected continuous chunks of the same task. Therefore, for each task, we select the longest continuous chunk as our model's predicted task segment. The extracted tasks are then used for performance metrics evaluation.

Efficiency Metrics Computation: Efficiency metrics are computed for each task identified from within the overall procedure using the models described in the previous section. The metrics are computed from robotic system data, such as joint angles (or kinematics) and button presses (events), specific to the da Vinci surgical system. System data was collected at 50Hz and synchronized to a single channel of endoscopic video.

The metrics used in this work are the same as those previously shown to be useful in differentiating surgoen experience and correlating to outcomes [67]. The metrics include Economy of Motion, Speed, Camera movements etc. Some examples of event- and kinematics- based metrics we evaluate are given in Table 7.1.

Efficiency Metrics Evaluation: In order to evaluate how well our procedure segmentation model works for the end goal of metrics evaluations, we compare the resulting efficiency metrics with those from human task labels (i.e., ground truth). A single, surgical resident manually identified the begin and end times for each surgical task using video review.

Table 7.1: Few examples of events and kinematics based metrics used for evaluation.

	Metric names
Event based	camera control on/off, energy on/off, master clutch on/off, head in/out, arm swap.
Kinematics based	economy of motion, master workspace range, wrist angles (roll, pitch, yaw), speed

Pearson's correlation coefficient between ground truth and ML-predicted metrics was used to evaluate model performance.

7.3 Dataset

We use the same dataset described in Chapter 6 for this work. However, we collect an additional 42 similar cases of RARP as a case study to see how well the procedure segmentation model can work in a real-time system for post-op performance report generation. This resulted in a total of 142 cases from which 62 (20 from previous chapter test set and 42 new) were used as a test set in this chapter.

7.4 Results and Discussion

Errors for the predicted begin and end boundaries of each surgical task are shown in Figure 7.2. We can see that certain tasks (like task 5 - Seminal Vesicles) have quite small errors whereas other tasks like task 12 have much larger errors. This could be a result of these different tasks having different variability across patients and technique. Some tasks might be very standardized and anatomical variations minimal whereas others require different approaches each time due to patient differences.

Looking at how well metrics were predicted, certain metrics (e.g economy of motion, camera control, etc) were very accurately predicted in most cases whereas others weren't. Table 7.4 shows the average correlation (over all metrics) of predicted vs ground truth. Event based metrics might be more sensitive to inaccuracies on begin/end boundaries than kinematic since events come in bursts - if you miss a burst, it can significantly influence the metric, and vice versa. Example scatter-plots for individual metrics are shown in Figure

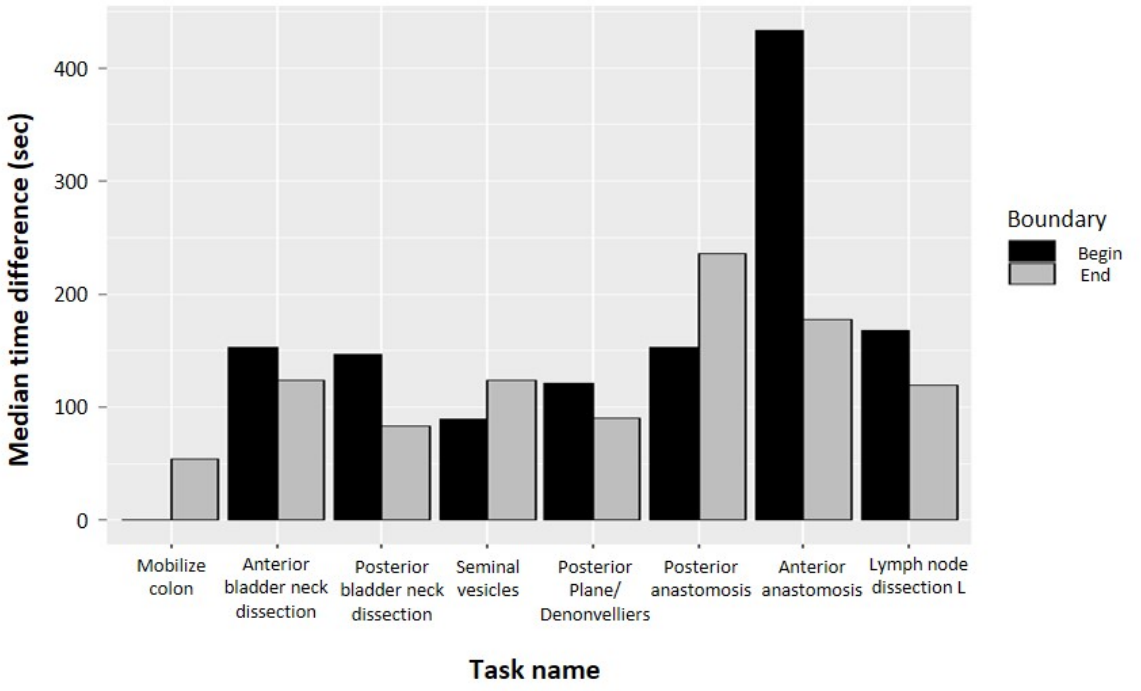

Figure 7.2: Results for difference in start and stop times of predictions vs ground truth. Each task has two box plots - one for start time and other for stop time.

7.3. In an ideal case, the values would fall on the unity line. Again, certain metrics follow this trend more than others. Importantly, not all tasks are of equal value for learning. There may be critical steps whose performance is more important. Some of the cardinal steps of prostatecomy include bladder neck dissection and anterior anastomosis. Our results for these steps illustrate that automated performance reports are feasible. In fact, some tasks may actually have less stringent requirements in terms of accurately predicted boundaries and based on estimated metrics than another task.

Since this work is aimed at developing a framework that can be used in a clinical setting, we also look at the processing times required on different kind of hardware. For this, we simulated how the data would be saved during an actual capture and timed procedure segmentation along with metric evaluations on three systems. The first one had a medium power CPU (without GPU), the second one had a single low end GPU (K2100M), and the last one had a single NVIDIA Titan X. The performance comparison is given in Figure

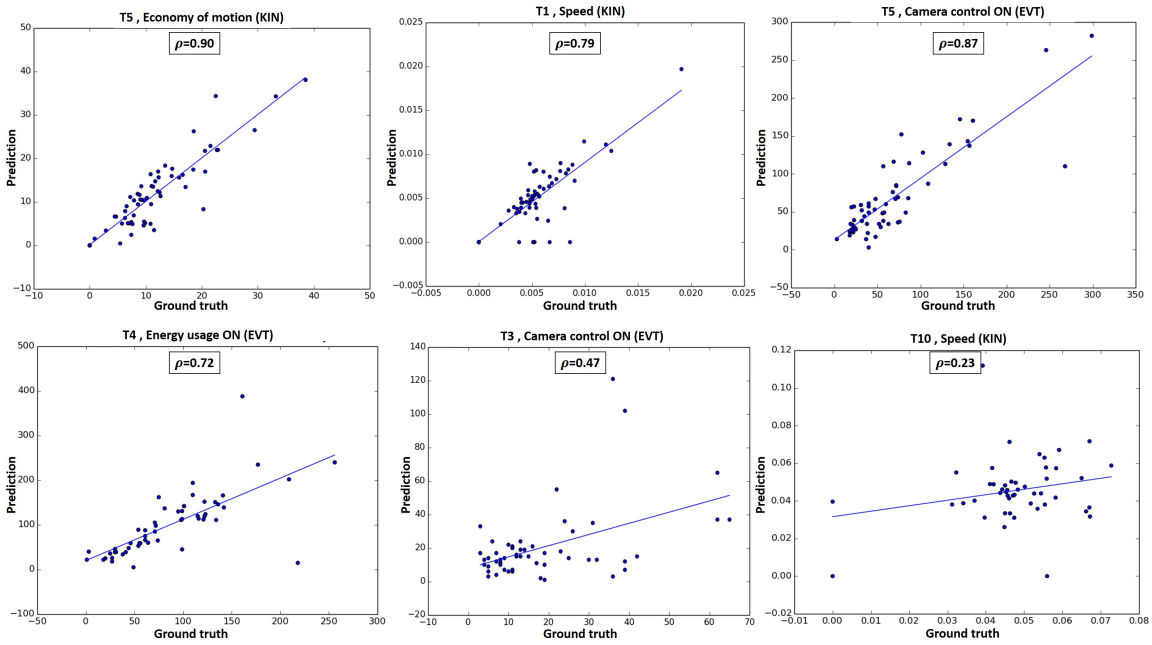

Figure 7.3: Scatter plots of different metrics. The x and y axis in each plot represent ground truth and predictions, respectively. The value of pearson correlation coefficient, task number and the name of corresponding metric is given on each plot.

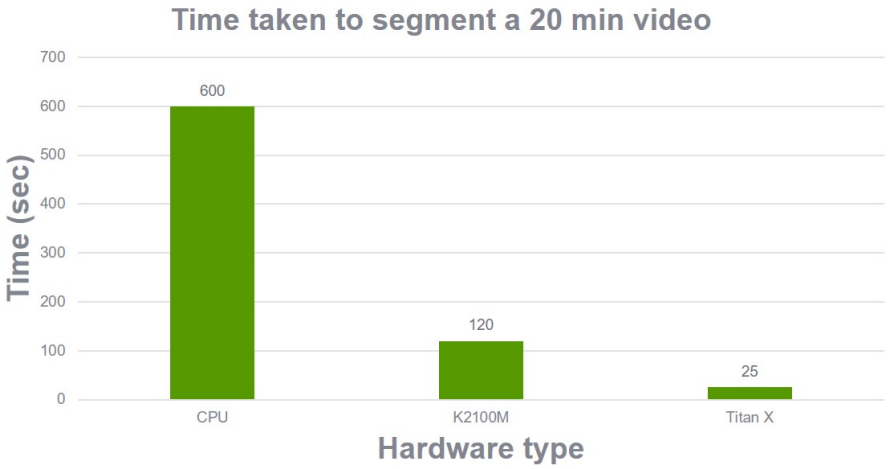

Figure 7.4: Processing time comparison of segmenting a 20 min video clip in raw form from the robot.

Table 7.2: Results for metric comparisons between ML predictions and ground truth. Each cell shows average pearson correlation coefficient over all metrics used for the corresponding task.

Task	Average Correlation Coefficient (EVT)	Average Correlation Coefficient (KIN)
Apical dissection	0.48 ± 0.27	0.127 ± 0.14
Posterior anastomosis	0.11 ± 0.13	0.128 ± 0.12
Anterior anastomosis	0.30 ± 0.27	0.352 ± 0.10
Posterior plane / Denonvilliers	0.34 ± 0.33	0.465 ± 0.13
Anterior bladder neck dissection	0.403 ± 0.3	0.424 ± 0.15
Lymph node dissection L	0.71 ± 0.16	0.446 ± 0.19
Posterior bladder neck dissection	0.650 ± 0.12	0.509 ± 0.15
mobilize colon / drop bladder	0.649 ± 0.22	0.706 ± 0.07
Seminal vesicles	0.781 ± 0.09	0.843 ± 0.11

7.4. As expected, the system with the best graphic card was the fastest. However, in our comparison of segmenting a 20 min video clip, we see that the worst performance of using a single CPU was still twice as fast as real time (just 600 sec - 10 min). This shows that even a non-GPU laptop can potentially provide surgeons with a report right after the procedure is completed.

While the results presented look very promising, some limitations also exist in this work. First, additional annotators could provide better consensus of ground truth which is used to train the ML models. Second, additional state information from the system could be used to improve model performance. One example could be instrument type. Third, metrics could be defined to be more robust to boundary errors. These were those used in prior work for relevance. Moreover, there is a need to test generalizability of models/metrics to other clinical sites (currently all data was from one site). And lastly, more data and continued model exploration could also help in improving performances.

7.5 Summary

This chapter proposed a new, applied method to evaluate ML models for procedure segmentation based on their impact to efficiency metrics in generating automated performance

reports. Given the presented results, it seems feasible that these reports can be automated, especially for a subset of critical tasks within an overall procedure.

CHAPTER 8

VIDEO HIGHLIGHTS FOR ROBOT-ASSISTED SURGERIES

All the previous chapters have dealt with giving surgeons better automated feedback in one form or the other. However, all the work presented in this thesis before had focused on evaluating surgical skill on a procedure or individual task level. Apart from giving feedback to surgeons in terms of skill score predictions on a whole task, it could be of great help to surgeons if they knew which parts of the task impacted their final score the most. The plot used to highlight impact over the course of a task will be called *'Task Highlights'*. This can potentially allow surgeons to focus more on specific techniques or gestures that contribute to low scores. Few works have presented approaches for measuring impact of different segments on overall skill score predictions. For example, in [68], the authors presented an approach using human pose for evaluating the impact of a particular segment on final score prediction in Olympic sports. We take inspiration from their work and in order to generate highlights in surgical tasks. However, similar to the work in [68], there are some key challenges in generating such highlights that are discussed below.

8.1 Challenges

1. **No ground truth:** There is no dataset available as of today that has annotated surgical tasks with individual window impact scores. Gathering any annotated surgical data set is hard by itself due to the busy schedules of experts; having such experts annotate each small section within surgical tasks is an even bigger ask. Therefore, none of the publicly available/self-collected datasets have such ground truth information available

2. **Difficult to validate results:** Similar to the problem of getting ground truth *'high-*

lights', there is also an issue of how the generated results can be validated. Once again, for this to happen, an expert has to watch the generated highlights along with the videos to approve/disapprove.

While these challenges exist and are hard to get around with, we still make an attempt at generating video highlights from surgical tasks and try to make sense of them in this chapter.

8.2 Dataset

We use JIGSAWS data set for our experiments in this chapter - details of JIGSAWS can be found in Chapter 4. We choose this data set since this is the only dataset available with gesture annotations that can help us understand the highlights generated better.

8.3 Methodology

We take the skill evaluation mechanism described in Chapter 4 and use it here as is. Support vector regression is used on features extracted from time series data to get final skill score (see Figure 4.1). We define the impact of a segment as the amount by which the predicted score would change if that segment was not observed. In order find the impact of segments within task, we use the DCT feature as described in Chapter 3 and 4. The goal here is to evaluate the inferred frequency feature vector had we not observed a particular segment of the data and then evaluate surgical score.

For a given d-th dimension of the kinematic time series $S(d) \in \Re^L$, the corresponding DCT features $F(d) \in \Re^L$ are evaluated using $F(d) = AS(d)$, where $A \in \Re^{L \times L}$ is the DCT transformation matrix. Taking $B = A^+$ as the inverse cosine transformation matrix (where A^+ denotes the pseudo-inverse of A), the DCT equation can be written as $F(d) = B^+S(d)$. Now, if the data from frames n_1 till n_2 were to be removed, we can evaluate the inferred DCT feature vector by $\hat{F}(d) = (B_{n_1:n_2})^+S(d)$, where $B_{n_1:n_2}$ is the matrix B with rows n_1

Table 8.1: Gesture vocabulary [48].

Gesture ID	Description
G1	Reaching for needle with right hand
G2	Positioning needle
G3	Pushing needle through tissue
G4	Transferring needle from left to right
G5	Moving to center with needle in grip
G6	Pulling suture with left hand
G7	Pulling suture with right hand
G8	Orienting needle
G9	Using right hand to help tighten suture
G10	Loosening more suture
G11	Dropping suture at end and moving to end points
G12	Reaching for needle with left hand
G13	Making C loop around right hand
G14	Reaching for suture with right hand
G15	Pulling suture with both hands

till n_2 removed. $\hat{F}$ will essentially have inferred the missing segment by the most likely kinematics signal given the frequency spectrum of the rest of the signal. Since $\hat{F}$ will have the same dimensionality as F, we can use the same SVR model for score prediction. The final impact of the segment on skill score is then evaluated by $impact = \psi - \hat{\psi}$, where ψ is the predicted score using whole sequence and $\hat{\psi}$ is the inferred score with a missing segment. The surgical task highlights can be generated by evaluating the $impact$ on a running window.

We use 50 lowest DCT features (same as for classification/prediction). The length of the running window had to be carefully selected in order to generate meaningful highlights. Having a long window length as compared to the length of the whole task would result in high impact scores for each segment since we might be omitting a significant part of skill relevant portion in a task, and vice versa. Therefore,after experimenting with different values of window lengths like 50, 100, 200 etc, we found a window length of 100 to work best for the dataset at hand for most cases.

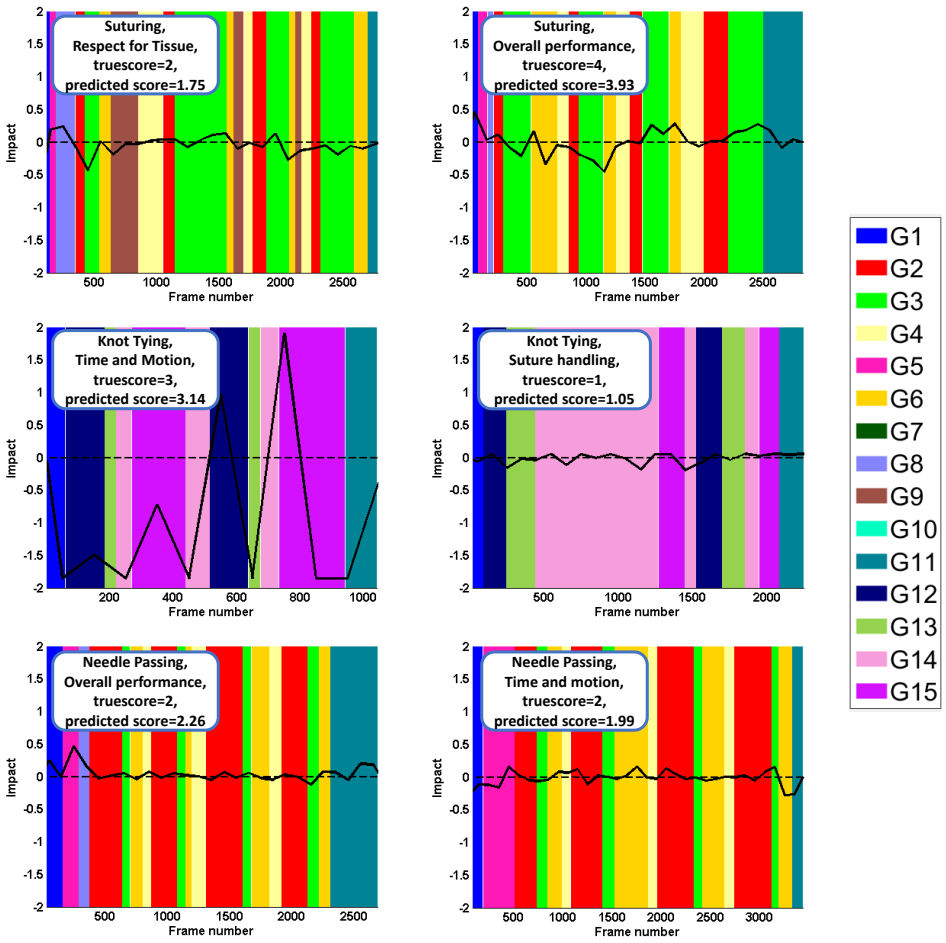

Figure 8.1: Sample task highlights. The y-axis on each plot corresponds to the impact (as defined in methodology section) with number of frames on the x-axis. The task type, modified-OSATS criteria, ground truth score, and the predicted score from our model using DCT features on the whole sequence, are given in boxes next to each plot. The color coding for the different gestures is also provided. The names of the gestures can be found in Table 8.1.

8.4 Results and Discussion

Figure 8.1 shows some sample task highlights constructed by following the procedure described in the methodology section. We overlay the impact scores plot on color coded gestures for getting better insights. The gestures used are the same as presented in the original dataset paper [48]. The gesture vocabulary of JIGSAWS dataset is given in Table 8.1.

The segments where the impact scores are negative indicate that these parts had a adverse effect on the final score, and vice versa. There are some interesting points that we can note from these plots that make intuitive sense. For example, in the suturing plot, we can observe that the impact score has maximum variations for G3 (i.e. Pushing needle through tissue). Since we predict for RT criteria in this case, one would expect that a 'good' or 'bad' push of a needle through the tissue should have the maximum impact on final skill score prediction. Similarly, for knot tying, we can see high positive and negative impact scores for G15 (i.e. Pulling suture with both hands). Again, this makes intuitive sense since G15 is important for knot tying task. We can draw similar insights for needle-passing (considering G2 and G5) as well. Although there are no ground-truth highlights to compare our results to (and it probably would be an extremely tedious task to generate such ground-truths), we believe that such impact score plots can tremendously help surgeons in understanding the parts within a task that they need to improve on. As a result, surgical trainees can direct their time and training on specific gestures within a task which can potentially allow them to move through their learning curves much faster.

8.5 Summary

In this Chapter, we explored an approach for generating video highlights in RMIS basic training. While there was no ground truth available in this case, our results seem to make sense when looked with overlaying surgical gestures. This work provides good basis for and provides direction to how such works can be extended to more complex clinical cases.

CHAPTER 9
SUMMARY AND FUTURE WORK

The main contributions of this thesis can be summarized as given below.

1. One of the largest studies on basic surgical skill assessment was carried out with data collected from 41 participants in total. Novel motion based features to differentiate surgical skill level were proposed that out-performed previous state-of-the-art models.

2. Novel framework and feature fusion approach for assessment in RMIS surgical training that out-performed all previous HMM based methods was proposed.

3. Novel unsupervised and supervised approaches were presented for procedure segmentation achieving high accuracies.

4. A large study was conducted on generating automated performance report on clinical robot-assisted radical prostatectomy. High performance achieved on predicting efficiency metrics makes the work possibly ready for implementation on a real system.

5. Novel approach presented for generating video highlights for giving surgeons more directed feedback in terms of which parts within a task effected their final score the most.

The promising results presented for automated benchmarking of surgical skills can be of huge benefit to the research and clinical community. However, there are still many things that can still be done to improve the presented work even further. Some of the future directions for the research work presented in this thesis are given below.

1. Inclusion of depth modality for assessment of basic suturing and knot tying skills as presented in Chapter 3. Having RGB-D can potentially help in extracting better motion features using video data.

2. Using pose instead of STIPs for extracting motion information from videos for human action assessment. STIPs make it hard to back track parts of a video which effected final score of a surgeon due to the clustering step. Using 2-D human pose will have most of the motion information needed for skill assessment and will allow for specific feedback to be given to surgeons in terms of how the hands can be moved for better scores.

3. Fusion of video and kinematics based features for RMIS training assessment. Using just kinematics data for assessment in RMIS made the work easy to deploy on a real system due to its light nature. However, in order to improve score prediction performance, using video based features can help tremendously as evidenced by success of video+accelerometer feature fusion in Chapter 3. With the availability of high power GPUs, the compute time may not be an issue.

4. Using deep learning based methods for surgical activity clustering. The methods presented in Chapter 5 worked great with a huge advantage of not requiring any training data as such. However, having more data can allow for powerful deep learning based unsupervised methods to be explored in this field.

5. Model improvements for procedure segmentation in RARP. There are many things that can be done to improve model performances for this problem. First, optical flow from videos can be used in a multi-stream CNN architecture. Second, LSTM based kinematics models can be combined with multi-image model. These are just few of many things that be done to improve recognition models.

6. Post-processing can be significantly improved in procedure segmentation. We pre-

sented the simplest of techniques using a median filter to de-noise segmentation output. However, many other things can be done in order to further improve this step. For example, a state machine based model can be implemented on top of median filtering to remove incorrect transitions from output.

7. Evaluation of APR work on data from different sites. Currently, all the data came from one site - in order to generalize well, data from different centers should be collected and tested upon.

8. Extension of APR work to other RA procedures. It would be interesting to see how models trained for RARP extend to other procedures like Hernia etc. There may not be a need for large amounts of data for new procedure as RARP procedure segmentation model weights could be used as initialization point.

9. Extension of video highlights work to clinical RA procedures. While RA procedures don't have ground truth like that of OSATS, scoring mechanisms can be developed using efficiency metrics which can then be used in a similar fashion to produce task wise video highlights for RA clinical cases.

REFERENCES

[1] R. Reznick and H. MacRae, "Teaching surgical skills–changes in the wind," *The New England journal of medicine*, vol. 355, no. 25, p. 2664, 2006.

[2] T. Yu, B. Wheeler, and A. Hill, "Clinical supervisor evaluations during general surgery clerkships," *Medical Teacher*, vol. 33, no. 9, pp. 479–484, 2011.

[3] J. Martin, G. Regehr, R. Reznick, H. MacRae, J. Murnaghan, C. Hutchison, and M. Brown, "Objective structured assessment of technical skill (osats) for surgical residents," *British Journal of Surgery*, vol. 84, no. 2, pp. 273–278, 1997.

[4] Y. Sharma, T. Plötz, N. Hammerla, S. Mellor, M. Roisin, P. Olivier, S. Deshmukh, A. McCaskie, and I. Essa, "Automated surgical OSATS prediction from videos," in *ISBI*, IEEE, 2014.

[5] Y. Sharma, V. Bettadapura, T. Plötz, N. Hammerla, S. Mellor, R. McNaney, P. Olivier, S. Deshmukh, A. McCaskie, and I. Essa, "Video based assessment of OSATS using sequential motion textures," in *International Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI)-Workshop*, 2014.

[6] V. Bettadapura, G. Schindler, T. Plötz, and I. Essa, "Augmenting bag-of-words: Data-driven discovery of temporal and structural information for activity recognition," in *CVPR*, IEEE, 2013.

[7] I. Nisky, Y. Che, Z. F. Quek, M. Weber, M. H. Hsieh, and A. M. Okamura, "Teleoperated versus open needle driving: Kinematic analysis of experienced surgeons and novice users," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 5371–5377.

[8] N. Ahmidi, Y. Gao, B. Béjar, S. S. Vedula, S. Khudanpur, R. Vidal, and G. D. Hager, "String motif-based description of tool motion for detecting skill and gestures in robotic surgery," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*, Springer, 2013, pp. 26–33.

[9] M. J. Fard, S. Ameri, R. B. Chinnam, A. K. Pandya, M. D. Klein, and R. D. Ellis, "Machine learning approach for skill evaluation in robotic-assisted surgery," *arXiv preprint arXiv:1611.05136*, 2016.

[10] J. Rosen, B. Hannaford, C. G. Richards, and M. N. Sinanan, "Markov modeling of minimally invasive surgery based on tool/tissue interaction and force/torque signa-

tures for evaluating surgical skills," *IEEE transactions on Biomedical Engineering*, vol. 48, no. 5, pp. 579–591, 2001.

[11] C. Reiley and G. Hager, "Decomposition of robotic surgical tasks: An analysis of subtasks and their correlation to skill," in *MICCAI*, 2009.

[12] L. Tao, L. Zappella, G. D. Hager, and R. Vidal, "Surgical gesture segmentation and recognition," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*, Springer, 2013, pp. 339–346.

[13] L. Tao, E. Elhamifar, S. Khudanpur, G. Hager, and R. Vidal, "Sparse hidden markov models for surgical gesture classification and skill evaluation," *Information Processing in Computer-Assisted Interventions*, pp. 167–177, 2012.

[14] A. C. Goh, D. W. Goldfarb, J. C. Sander, B. J. Miles, and B. J. Dunkin, "Global evaluative assessment of robotic skills: Validation of a clinical assessment tool to measure robotic surgical skills," *The Journal of urology*, vol. 187, no. 1, pp. 247–252, 2012.

[15] M. Ershad, Z. Koesters, R. Rege, and A. Majewicz, "Meaningful assessment of surgical expertise: Semantic labeling with data and crowds," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer International Publishing, 2016, pp. 508–515.

[16] R. DiPietro, C. Lea, A. Malpani, N. Ahmidi, S. S. Vedula, G. I. Lee, M. R. Lee, and G. D. Hager, "Recognizing surgical activities with recurrent neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer International Publishing, 2016, pp. 551–558.

[17] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, "Endonet: A deep architecture for recognition tasks on laparoscopic videos," *arXiv preprint arXiv:1602.03012*, 2016.

[18] C. Lea, G. D. Hager, and R. Vidal, "An improved model for segmentation and recognition of fine-grained activities with application to surgical training tasks," in *2015 IEEE Winter Conference on Applications of Computer Vision*, IEEE, 2015, pp. 1123–1129.

[19] B. B. Haro, L. Zappella, and R. Vidal, "Surgical gesture classification from video data," in *MICCAI 2012*, Springer, 2012, pp. 34–41.

[20] L. Zappella, B. Béjar, G. Hager, and R. Vidal, "Surgical gesture classification from video and kinematic data," *Medical Image Analysis*, vol. 17, no. 7, 732–745, 2013.

[21] N. Padoy, T. Blum, S.-A. Ahmadi, H. Feussner, M.-O. Berger, and N. Navab, "Statistical modeling and recognition of surgical workflow," *Medical Image Analysis*, vol. 16, no. 3, pp. 632–641, 2012.

[22] F. Lalys, L. Riffaud, D. Bouget, and P. Jannin, "An application-dependent framework for the recognition of high-level surgical tasks in the or," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, Springer, 2011, pp. 331–338.

[23] T. Blum, H. Feußner, and N. Navab, "Modeling and segmentation of surgical workflow from laparoscopic video," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*, Springer, 2010, pp. 400–407.

[24] H. Lin and G. Hager, "User-independent models of manipulation using video contextual cues," in *International Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI)-Workshop*, 2009.

[25] H. Lin, I. Shafran, T. Murphy, A. Okamura, D. Yuh, and G. Hager, "Automatic detection and segmentation of robot-assisted surgical motions," *MICCAI*, 2005.

[26] C. Reiley, H. Lin, B. Varadarajan, B. Vagvolgyi, S. Khudanpur, D. Yuh, and G. Hager, "Automatic recognition of surgical motions using statistical modeling for capturing variability," *Studies in health technology and informatics*, vol. 132, p. 396, 2008.

[27] G. Saggio, G. Santosuosso, P. Cavallo, C. Pinto, M. Petrella, F. Giannini, N. Di Lorenzo, A. Lazzaro, A. Corona, F. D'Auria, *et al.*, "Gesture recognition and classification for surgical skill assessment," in *MeMeA*, IEEE, 2011, pp. 662–666.

[28] M. Sahu, A. Mukhopadhyay, A. Szengel, and S. Zachow, "Addressing multi-label imbalance problem of surgical tool detection using cnn," *International journal of computer assisted radiology and surgery*, vol. 12, no. 6, pp. 1013–1020, 2017.

[29] D. Katić, C. Julliard, A.-L. Wekerle, H. Kenngott, B. P. Müller-Stich, R. Dillmann, S. Speidel, P. Jannin, and B. Gibaud, "Lapontospm: An ontology for laparoscopic surgeries and its application to surgical phase recognition," *International journal of computer assisted radiology and surgery*, vol. 10, no. 9, pp. 1427–1434, 2015.

[30] O. Dergachyova, D. Bouget, A. Huaulmé, X. Morandi, and P. Jannin, "Automatic data-driven real-time segmentation and recognition of surgical workflow," *International journal of computer assisted radiology and surgery*, vol. 11, no. 6, pp. 1081–1089, 2016.

[31] S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg, "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learn-

ing," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1595–1618, 2017.

[32] F. Despinoy, D. Bouget, G. Forestier, C. Penet, N. Zemiti, P. Poignet, and P. Jannin, "Unsupervised trajectory segmentation for surgical gesture recognition in robotic training," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1280–1291, 2016.

[33] A. Zia, D. Castro, and I. Essa, "Fine-tuning deep architectures for surgical tool detection,"

[34] M. Sahu, A. Mukhopadhyay, A. Szengel, and S. Zachow, "Tool and phase recognition using contextual cnn features," *arXiv preprint arXiv:1610.08854*, 2016.

[35] N. Ahmidi, P. Poddar, J. D. Jones, S. S. Vedula, L. Ishii, G. D. Hager, and M. Ishii, "Automated objective surgical skill assessment in the operating room from unstructured tool motion in septoplasty," *International journal of computer assisted radiology and surgery*, vol. 10, no. 6, pp. 981–991, 2015.

[36] F. Lalys, D. Bouget, L. Riffaud, and P. Jannin, "Automatic knowledge-based recognition of low-level tasks in ophthalmological procedures," *International journal of computer assisted radiology and surgery*, vol. 8, no. 1, pp. 39–49, 2013.

[37] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, M. A. Clements, and I. Essa, "Automated assessment of surgical skills using frequency analysis," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, Springer, 2015, pp. 430–438.

[38] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, T. Ploetz, M. A. Clements, and I. Essa, "Automated video-based assessment of surgical skills for training and evaluation in medical schools," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 9, pp. 1623–1636, 2016.

[39] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, and I. Essa, "Video and accelerometer-based motion analysis for automated surgical skills assessment," *International journal of computer assisted radiology and surgery*, vol. 13, no. 3, pp. 443–455, 2018.

[40] I. Laptev and T. Lindeberg, "Space-time interest points," in *IN ICCV*, 2003, pp. 432–439.

[41] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.

[42] S. M. Pincus, "Approximate entropy as a measure of system complexity.," *Proceedings of the National Academy of Sciences*, vol. 88, no. 6, pp. 2297–2301, 1991.

[43] S. Pincus and B. H. Singer, "Randomness and degrees of irregularity," *Proceedings of the National Academy of Sciences*, vol. 93, no. 5, pp. 2083–2088, 1996.

[44] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern recognition letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[45] H. Sloetjes and P. Wittenburg, "Annotation by category: Elan and iso dcr.," in *LREC*, 2008.

[46] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[47] M. Martínez-Zarzuela, C. Gómez, F. J. D. Pernas, A. Fernández, and R. Hornero, "Cross-approximate entropy parallel computation on gpus for biomedical signal analysis. application to meg recordings," *Computer Methods and Programs in Biomedicine*, vol. 112, pp. 189–199, 2013.

[48] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh, C. C. G. Chen, R. Vidal, S. Khundanpur, and G. D. Hager, "Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling," in *MICCAI Workshop: M2CAI*, vol. 3, 2014.

[49] A. Zia and I. Essa, "Automated surgical skill assessment in rmis training," *International journal of computer assisted radiology and surgery*, vol. 13, no. 5, pp. 731–739, 2018.

[50] L. Tao, E. Elhamifar, S. Khudanpur, G. D. Hager, and R. Vidal, "Sparse hidden markov models for surgical gesture classification and skill evaluation," in *International Conference on Information Processing in Computer-Assisted Interventions*, Springer Berlin Heidelberg, 2012, pp. 167–177.

[51] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems 9*, M. I. Jordan and T. Petsche, Eds., MIT Press, 1997, pp. 155–161.

[52] A. Zia, C. Zhang, X. Xiong, and A. M. Jarc, "Temporal clustering of surgical activities in robot-assisted surgery," *International journal of computer assisted radiology and surgery*, vol. 12, no. 7, pp. 1171–1178, 2017.

[53] F. Zhou, F. De la Torre, and J. K. Hodgins, "Aligned cluster analysis for temporal segmentation of human motion," in *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, IEEE, 2008, pp. 1–7.

[54] ——, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *PAMI*, vol. 35, no. 3, 2013.

[55] F. Wang and C. Zhang, "Spectral clustering for time series," in *Pattern Recognition and Data Mining: Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Bath, UK, August 22-25, 2005, Proceedings, Part I*, S. Singh, M. Singh, C. Apte, and P. Perner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 345–354, ISBN: 978-3-540-28758-2.

[56] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, "Segmenting motion capture data into distinct behaviors," in *Proceedings of Graphics Interface 2004*, Canadian Human-Computer Communications Society, 2004, pp. 185–194.

[57] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[58] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, "Endonet: A deep architecture for recognition tasks on laparoscopic videos," *CoRR*, vol. abs/1602.03012, 2016.

[59] A. Zia, A. Hung, I. Essa, and A. Jarc, "Surgical activity recognition in robot-assisted radical prostatectomy using deep learning," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Springer International Publishing, 2018, pp. 273–280.

[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[62] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[63] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Computer Vision (ICCV), 2015 IEEE International Conference on*, IEEE, 2015, pp. 4489–4497.

[64] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, pp. 248–255.

[65] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[66] A. J. Hung, J. Chen, A. Jarc, D. Hatcher, H. Djaladat, and I. S. Gill, "Development and validation of objective performance metrics for robot-assisted radical prostatectomy: A pilot study," *The Journal of urology*, vol. 199, no. 1, pp. 296–304, 2018.

[67] A. J. Hung, J. Chen, Z. Che, T. Nilanon, A. Jarc, M. Titus, P. J. Oh, I. S. Gill, and Y. Liu, "Utilizing machine learning and automated performance metrics to evaluate robot-assisted radical prostatectomy performance and predict outcomes," *Journal of endourology*, vol. 32, no. 5, pp. 438–444, 2018.

[68] H. Pirsiavash, C. Vondrick, and A. Torralba, "Assessing the quality of actions," in *European Conference on Computer Vision*, Springer International Publishing, 2014, pp. 556–571.