Prediction of laparoscopic cholecystectomy procedure duration using artificial intelligence

Nicolas Eleni van de Kar

MASTER THESIS

Prediction of laparoscopic cholecystectomy procedure duration using artificial intelligence

Author:
Nicolas Eleni van de Kar, BSc

A thesis submitted in fulfilment of the requirements
for the degree of Master of Science
in
TECHNICAL MEDICINE

MEANDER MEDICAL CENTER
Department of Surgery & Center for AI

UNIVERSITY OF TWENTE
Faculty of Science and Technology

01-03-2022

Preface

This thesis is submitted in fulfilment of the requirements for the degree of Master of Science (MSc) in Technical Medicine at the University of Twente. The thesis describes the research conducted during my graduation internship at the Meander Medical Centre in Amersfoort. The field of medical data-analysis and artificial intelligence has caught my interest during the master. There were two aspects of this graduation position which motivated me to choose for this hospital. The first was the application of artificial intelligence on surgical data and secondly the collaboration of the Meander Medical Centre with a multinational MedTech company. The AI-Lab was founded based on the mutual interest of using artificial intelligence to assist, evaluate and optimize surgical processes. For my internship the subject of optimizing the operating room process was of interest. The application of artificial intelligence on a high volume surgery, laparoscopic cholecystectomy, was selected to evaluate a proof of concept. The large amount of video data makes this procedure suited to apply artificial intelligence on. This has led to the subject of my thesis: Prediction of laparoscopic cholecystectomy procedure duration using artificial intelligence.

In the period of this internship I was surrounded by great group of intelligent people, who supervised and collaborated with me on this research. I would like to give my appreciation to Prof. Dr. Broeders for creating the ideal environment to develop myself as a young professional in the MedTech field, give motivation and advice. I want to thank Dr. ir. Ferdi van der Heijden and Dr. Can Tan for the technical supervision during my internship(s). Our meetings helped me explore the field of artificial intelligence and triggered me to have a better understanding of the difficult technical aspects of scientific research. I would like to express my gratitude to Julian Abbing MSc for his daily supervision, technical support and helped me develop skills for future careers. I cannot thank you enough for that. I also want to extend my appreciation to Bregje Hessink-Sweep MSc. Even though, we have not seen each other that much in person due to COVID. In the online sessions, you provided me with carefully thought through tips, feedback, and insights. I have learned much about myself, which helped me shape into the person I am today. I want to thank the PhD- and TM-students at the Meander Medical Centre for the great collaborations, support and nice coffee / lunch breaks. I want to give my special thanks to Christianne. Your constant loving support helped me not only to complete this thesis but through-out my hole studies. Finally, I want to thank my parents for the constant support and advice over the years.

Abstract

A cholecystectomy is the procedure of the surgical removal of a diseased gallbladder. Each year, more than 25,000 cholecystectomies are performed by surgeons in the Netherlands. The high volume of the procedure makes it suited for artificial intelligence applications. The aim of this study is the development of an artificial intelligence network that predicts the remaining procedure time for the laparoscopic cholecystectomy based on video data, and updates the estimated remaining procedure time during the procedure based on the progress.

The study consists of two parts. The first part is the development a deep learning network that can accurately and objectively classifying the surgical phases of intraoperative laparoscopic cholecystectomy (LC) videos. All 80 LC videos of the publicly available Cholec80 dataset are used as data source, for comparability with other studies. A residual neural network is used as a base-line deep learning network to classify the surgical phases. The classification results are post-processed by a moving window to filter the network output. After classification, the duration of the individual phases is extracted by detecting the phase transitions. In addition, the importance of adequate labelling of surgical video data is investigated. The network performance metrics of the original annotations of the Cholec80 dataset are compared with revised phase annotations, that are defined based on clinical relevance and technical capabilities. The second part consists of the prediction of the remaining procedure time after each surgical phase. The predictions are based on the phase duration, derived from the detected phase transitions by the phase detector. The model performance of linear regression, random-forest regression and support vector regression are evaluated for predicting the remaining procedure time.

The residual neural network has a 79.0% accuracy, 80.5% precision, 78.1% recall and 79.3% F1-score for the original annotations and 85.0% accuracy, 86.3% precision, 84.3% recall and 85.3% F1-score for the revised annotations on the test set. The revised annotation performance metrics showed an improvement of 6.0%, 5.8%, 6.2% and 6.0%, for accuracy, precision, recall and F1-score respectively compared to the original. Post-processing of the phase output removed the noisy character but was susceptible to artifacts. TCNs are advised for future research. The regression model accurately predicted the remaining procedure time based on the phase durations of the LC procedure. The random-forest regression model showed to be the best model to predict the remaining procedure time, with an overall RMSE of 8.5 min and R^2 of 0.6 on the test set and with a significant difference to almost all linear and support vector regression results. Although these results improve upon the performance stated in previous research, the model did not yield results that are within the defined standards for use in clinical practice. However, further improvements on the network, dataset and learning process, as described in the recommendations, might enable the possibility for clinical implementation.

Graduation Committee

Chairman & Medical Supervisor Institution
Prof. dr. I.A.M.J. Broeders, MD
Department of Surgery, Meander Medical Centre, Amersfoort
Robotics and Mechatronics, University of Twente, Enschede

Technical Supervisor University
Dr. C.O. Tan
Robotics and Mechatronics, University of Twente, Enschede

Technical Supervisor Institution
J.R. Abbing, MSc & PhD candidate
Department of Surgery, Meander Medical Centre, Amersfoort
Robotics and Mechatronics, University of Twente, Enschede

Process Supervisor University
A.G. Lovink, MSc
Technical Medicine, University of Twente, Enschede

External Member University
L. Molenaar, MSc & PhD candidate
Magnetic Detection & Imaging, University of Twente, Enschede

Table of Contents

List of Abbreviations

AC	**A**bdominal **C**avity
Adam	**Ada**ptive **m**oment estimation
AI	**A**rtificial **I**ntelligence
CA	**C**ystic **A**rtery
CAM	**C**lass **A**ctivation **M**ap
CART	**C**lassification **A**nd **R**egression **T**rees
CD	**C**ystic **D**uct
CE	**C**ross-**E**ntropy
CI	**C**onfidence **I**nterval
CM	**C**onfusion **M**atrix
CNN	**C**onvolutional **N**eural **N**etwork
CVRP	**C**omputer **V**ision and **P**attern **R**ecognition
CVS	**C**ritical **V**iew of **S**afety
DL	**D**eep **L**earning
DT	**D**ecision **T**rees
EAES	**E**uropean **A**ssociation for **E**ndoscopic **S**urgery
ECG	**E**cho**c**ardio**g**ram
EHR	**E**lectronic **H**ealth **R**ecord
ETA	**E**xpected **T**ime of **A**rrival
FC	**F**ully **C**onnected
FE	**F**eature **E**xtraction
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
fps	**f**rames **p**er **s**econd
GAP	**G**lobal **A**verage **P**ooling
GB	**G**all**b**ladder
GPU	**G**raphical **P**rocessing **U**nit
HMM	**H**idden **M**arkov **M**odel
LC	**L**aparoscopic **C**holecystectomy
LSTM	**L**ong-**S**hort **T**ime **M**emory
LR	**L**inear **R**egression
MAE	**M**ean **A**bsolute **E**rror
MAPE	**M**ean **A**bsolute **P**ercentage **E**rror
ML	**M**achine **L**earning
MLR	**M**ultiple **L**inear **R**egression
MMC	**M**eander **M**edical **C**entre
MSE	**M**ean **S**quared **E**rror
OR	**O**perating **R**oom
ORs	**O**perating **R**ooms
PD	**P**hase **D**etector
PIL	**P**ython **I**mage **L**ibrary
R^2	Coefficient of determination
RBF	**R**adial **B**asis **F**unction
ReLU	**Re**ctified **L**inear activation **U**nit
ResNet	**Res**idual Neural **Net**work
RF	**R**andom **F**orest
RGB	**R**ed **G**reen **B**lue

RMSprop	**R**oot **M**ean **S**quared **prop**agation
RMSE	**R**oot **M**ean **S**quared **E**rror
RNN	**R**ecurrent **N**eural **N**etwork
RPT	**R**emaining **P**rocedure **T**ime
SGD	**S**tochastic **G**radient **D**escent
SLR	**S**imple **L**inear **R**egression
std	**st**andard **d**eviation
SVM	**S**upport **V**ector **M**achine
SVR	**S**upport **V**ector **R**egression
TBC	**T**u**b**er**c**ulosis
TCN	**T**emporal **C**onvolutional **N**etwork
TN	**T**rue **N**egative
TP	**T**rue **P**ositive

CHAPTER 1

1. Introduction

This chapter discusses the clinical background for laparoscopic cholecystectomy surgery and the operating room scheduling process. An overview of previous studies into the application of artificial intelligence in healthcare and specifically for laparoscopic cholecystectomy surgery is presented. Based on this information, the clinical problem, research questions and the aim of this study are defined.

1.1 Clinical background

1.1.1 Laparoscopic Cholecystectomy procedure

Each year, more than 25,000 cholecystectomies are performed by surgeons in the Netherlands.[1] A cholecystectomy is the procedure of the surgical removal of a diseased gallbladder (GB). Indications for a cholecystectomy are acute or chronic cholecystitis, cholelithiasis, gallstone pancreatitis, biliary dyskinesia and GB masses or polyps.[2] Laparoscopic cholecystectomy (LC) is currently the gold standard for routine GB removal surgery. Since the early 1990s, LC essentially replaced the open surgery approach because of a decreased morbidity rate, shorter post-operative hospitalisation and faster recovery due to the minimal invasiveness.[3]

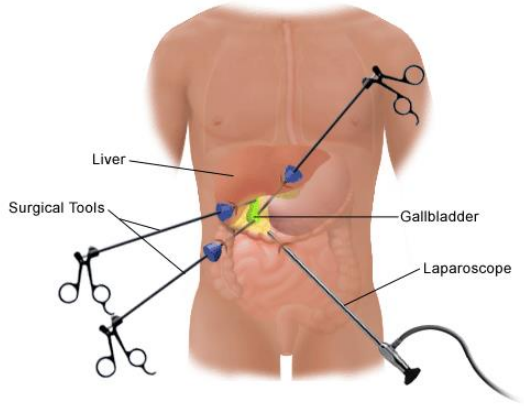

Figure 1.1: The port locations for a standard LC procedure.[4]

The standard technique to perform a LC uses four ports, three for the surgical tools and one for the laparoscope. First, a pneumoperitoneum is created. In most cases the closed Veress needle technique is applied, however a blunt or Hasson's trocar can also be used. The trocar for the laparoscope is placed either intra-, infra- or supraumbilical, depending on patient's body shape and preference of the surgeon. The three

trocars for the surgical tools are placed in the subxiphoid, lateral subxiphoid and medial subcostal port. The location of the four ports is shown in figure 1.1.

The steps of the LC procedure are shown in the images of figure 1.2 and will be discussed in detail. After the ports are placed, the liver is elevated with the surgical graspers to expose the GB that lies underneath. The elevation of the liver provides an overview of the gallbladder and the surrounding structures. Next, the fundus of the GB is elevated to take over the support of the liver and extend the GB. Hartmann's pouch is retracted for optimal visibility of the bile ducts and arteries. After creating optimal visibility, the dissection of Calot's triangle is performed to clear overlaying fat tissue and peritoneum of the cystic duct and artery. This dissection provides the Critical View of Safety (CVS), that is used to identify the critical structures prior to the transection.[5] The cystic duct (CD) and cystic artery (CA) are clipped proximal and distal to the location of the transection. The clips prevent blood loss by bleeding of the cystic artery, leakage of bile and the possible lost of gallstones. Leakage of bile and lost of gallstones in the abdominal cavity (AC) increase the risk of complications such as intraperitoneal abscesses and fistulas.[6-9] The cystic duct and artery are transected between the clips with scissors. After the transection, the dissection of the GB is performed by separating the GB from the liver. For the retraction from the AC, the GB is packaged in an extraction bag. The bag prevents the leakage of bile and lost of gallstones in the AC when the clips release due the increased force of the retraction through the abdominal wall. When the GB is packed, the trocar of one port is removed and the incision is stretched to provide for enough space to retract the GB.[10]

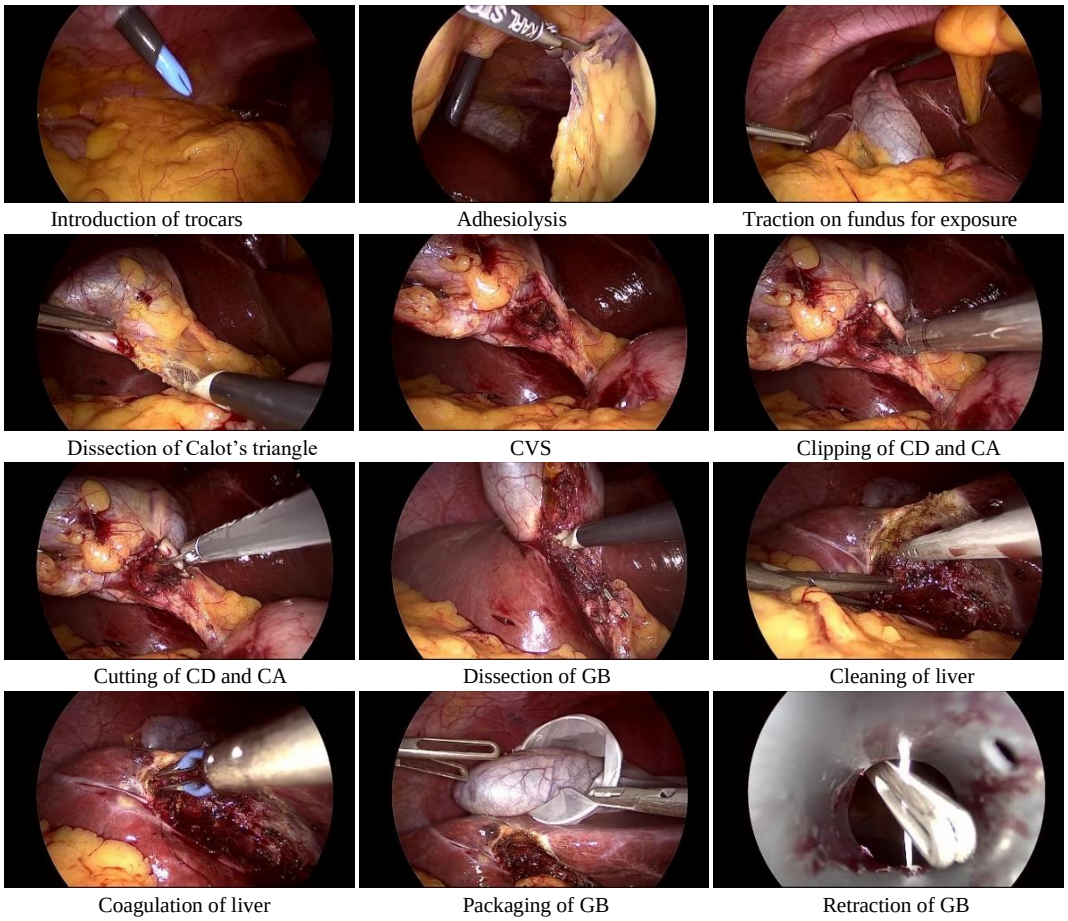

Figure 1.2: The surgical steps of the LC procedure.

1.1.2 Difficulties of a laparoscopic cholecystectomy

Although the LC procedure has evolved to a relatively safe operation, it does have some challenges. One of the difficulties of the LC is the increased hand-eye coordination needed by the surgeon, in comparison to an open surgery, in order to compensate for the indirect vision from a screen when performing surgical tasks. The LC procedures additionally requires skills to manually compensate for the amplification of errors in movement by the long surgical instruments. A third difficulty is related to the Fulcrum effect caused by the abdominal wall. This effect describes the opposite moment of the surgeon's hand outside and the tip of the instrument inside the AC. Finally, the surgeon needs to compensate for the lack of sensing with the surgical instruments and the lack of depth in the 2D laparoscopic videos.[12]

In order to overcome the difficult aspects of the LC, surgical training is needed. The surgical resident performs the surgery under direct supervision of an experienced surgeon. As the expertise of the resident develops, the level of supervision reduces. The learning curve of the resident is directly related to the training. The European Association for Endoscopic Surgery (EAES) guidelines indicate that a surgical resident needs to preform between 20-35 LC in order to operate safely without supervision.[13] An experienced surgeon is expected to be able to perform a LC in less than 60 minutes, junior surgeons show a significantly increase in operation time. Research showed that a prolonged operation time increases the risk of complications and a prolonged post-operative recovery.[14] The surgical experience level for the LC procedure is divided in three categories: inexperienced with less than ten LCs, intermediate between 20-50 LCs and experienced with over 100 LCs.[15]

Several studies examined the risk of the steps in the LC procedure.[6-9] Four steps showed an increased risk. First, the traction on the fundus and Hartmann's pouch with the gasper, in order to expose the GB for dissection of Calot's triangle, can lead to a rupture and possible lost of gallstones or leakage of bile. Second, the dissection and transection of the CD can lead to bile duct injury as a result of damage. In this case, the bile duct will not be able to function properly and bile might leak into the abdomen or the flow of bile from the liver is blocked. Third, the dissection of the GB from the liver can lead to a rupture or puncture and possible lost of gallstones or leakage of bile. Fourth and finally, the extraction of the GB through the abdominal wall poses a risk to the lost of gallstones and leakage of bile, when no bag is used for the extraction.[9] Therefore, the surgeons of the Meander Medical Center (MMC) always use a retrieval bag. The section above outlines the LC procedure and the (potential) difficulties surgeons encounter. Another aspect that is important to provide a proper environment to conduct LC and other procedures, is related to Operating Rooms (ORs) and Operating Room Planning.

1.2 Planning of operating rooms for surgical procedures

ORs are of great importance for medical centers because they provide the main revenue. They are, however, a large part of the costs as well.[16] The operating room (OR) planning of surgical procedures is known for its complexity. There are many factors that have to be taken into account, such as the availability of OR personnel and surgeons, constraints imposed by limited OR facilities, emergency procedures, and the large diversity of patients and procedures.[17 18] Another key element in OR planning is the duration of surgical procedures.[16-19] In current clinical practice, the preoperative predicted surgery duration is based on average durations and rough estimations. However, there is a large variability in duration observed for many

surgical procedures leading to suboptimal OR planning.[18-21] Surgical procedures that take longer than the expected operation time, induce a delay or even cancellation of subsequent procedures. As a result, patients experience longer waiting times and OR personnel has to work overtime. The increased preoperative waiting time leads to patient discomfort and might even pose a higher risk for complications. On the contrary, surgical procedures that finish prior to the expected operation time cause unnecessary vacancy of the OR. In order to adjust for the variability in surgery duration, the OR schedulers monitor the duration of each OR either by observation or verbal communication with the OR teams. The OR schedulers estimate the remaining procedure time case-by-case to adapt the OR schedule accordingly. The accuracy of this workflow relies on the extensive experience with various procedures and the estimation of the OR teams. Robust schedules require procedure duration estimations that are unbiased, accurate, and minimises cases with absolute errors.[22]

Improvements in the current practice of OR scheduling are automated systems that give real-time updates about the progress of the procedure and the capability of making reliable predictions of the procedure duration. These predictions could reduce the preoperative waiting time for patient, which improves the patient comfort and might also reduce patient risks. The automated retrieval of the progress information reduces the added registration burden on the OR team or interruption of the surgical process for communication.[23 24] The technologies that are used in the OR offer a source of information for an automated system. Specifically for laparoscopic procedures, as the LC, video data is a valuable source because it contains information about anatomical structures and the use of surgical tools.[24 25-27] An experienced observer can give a progress indication of the procedure based on the information of the laparoscopic videos. A computer algorithm should, in theory, also be able to retrieve the visual information about the progress of the procedure. These algorithms often use artificial intelligence (AI) to extract the information from the data.

1.3 Artificial intelligence

1.3.1 Artificial intelligence in healthcare

The constant strive to improve patient outcomes in healthcare and to lower the cost, request the development and introduction of new innovations. After the introduction of the digitisation in healthcare, applications for big data driven technologies as AI are researched extensively. AI is part of computer science which tries to make complex algorithms and machines that mimic cognitive characteristics of humans. AI is used in a wide range of applications such as the automotive industry, finance and smart devices. In medicine, AI is applied for automatic diagnostics, improved detection of pathologies and clinical decision making. The first clinical applications show great value in the detection of nodes, tuberculosis (TBC) and COVID in X-ray images, arrhythmias in echocardiograms (ECGs) and outcome prediction in infectious diseases. Application of AI by Google for automatic lung cancer detection is shown in figure 1.3.[28] The subfields of AI on which the current attention is focussed are machine learning (ML) and deep learning (DL). ML tries to find correlations and associations on predefined features in data. DL is again a subfield of ML, which uses an infrastructure which mimics the human brain called artificial neural networks. The network structure consists of numerous artificial ‘neurons’ stacked on each other in layers,

creating a deep neural network. The neural networks can be trained to define their own features in order to find correlations and associations in the data. The latter can be used for making predictions or classifications. The network's training is achieved by learning specific features from prelabelled data.[29]

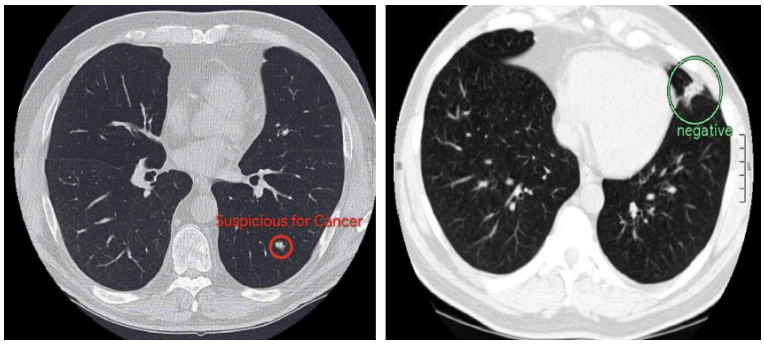

Figure 1.3: Google's lung cancer detection AI indicating suspicious and negative nodes for cancer.[28]

1.3.2 Centre for Artificial intelligence in Meander Medical Center

In the past five years, some research for the application of AI on clinical problems in surgery has been conducted in the MMC. The MMC has an extensive collaboration with Johnson & Johnson, and was previously working together with Verb Surgical for the development of a surgical robot. The MMC signed an alpha partner agreement with Johnson & Johnson for the research and development of digital solutions for surgery. Since December 2020, the 'Center for Artificial Intelligence' has been established in the MMC. The purpose of this centre is to create a platform that supports AI projects and enables the exchange of clinical data and research results. The center combines the interests of the MMC and Johnson & Johnson to develop innovative digital solutions for surgery. Johnson & Johnson gives technical support for these AI projects and the MMC provides the facilities and clinical data for the studies.

The application of AI for radiology purposes has been investigated extensively. This research led to the development of new products that are implemented in a wide range of applications within the radiology department. The application of AI for surgical purposes has, however, only been investigated marginally. The projects of the centre for AI explore the possibilities for clinical applications in surgery, mainly in laparoscopic cholecystectomies, totally extraperitoneal hernioplasty and fundoplication procedures. These projects consist of objectifying the performance of surgeons, to give more insight into and assist in the further improvement of their personal performance. This objectification is used to create a benchmark for surgical performance that can help surgical residents. Another project focusses on the assessment of intraoperative decision making with AI networks and give feedback on surgical performance. The last topic is the identification of anatomical structures and phase recognition of surgical procedures from laparoscopic videos.

1.3.3 Artificial intelligence for laparoscopic cholecystectomy

In the recent years, an increasing amount of papers are published on the application of AI on LC data. The interest in LCs originates from the fact that it is a high-volume surgical procedure, resulting in large datasets. The largest and most commonly used dataset is the publicly available Cholec80.[30] This dataset contains 80 LC videos that are annotated for the surgical phase and instruments. The dataset is used in studies for education, benchmarking, risk assessment and the prediction of remaining surgery time. Most

studies focused on improving of the results presented in previous studies about phase and instrument recognition. These two tasks are essential components for the objective assessment of surgical skills. Benchmarking of the surgical skills for surgeons proved to increase their performance.[31] The surgical skills are measured by analysing the order and duration of surgical steps, the type of instruments, the time instruments are used, the path length of instruments and the smoothness in movements.[31 32] The evaluation of these objective parameters improves the learning process, in particular for junior surgeons. This type of assessment enables personalised training, feedback based on skill level and objective surgery evaluation.[31]

The expansion of DL networks to medicine requires an increase in expertise and knowledge for adequate annotation processes. This is particularly the case as even the opinions of experts differ on annotation definitions. The networks used in medical applications often apply pattern recognition for tasks as surgical phase recognition, having a very high annotation difficulty. Most studies in surgical phase recognition focus solely on the development of a network structure with improved performance. The network performance and structure are, however, equally important as the data acquisition and data quality.[32 33] The latter raises the importance of a generalised annotation process, that incorporates both medical and technical expertise for adequate datasets used to train networks.[35]

One of the subjects for the implementation of AI in LC procedures is automatic difficulty grading of the procedure and the detection of bile leakage. Bile leakage and lost stones increase the risk of postoperative complications as the formation of abscesses and fistulas in the peritoneal cavity. The main problem is the missing report of gallbladder leakage, ranging from 13 - 78%. The network can detect bile and gallstone based on colour-based-feature-extraction with an accuracy of 83%.[36]

A promising application of AI in LC is surgical phase recognition. Extensive research in automatic recognition of surgical phases has led to investigation the application of this information for the prediction of remaining surgery time. This information can be used to improve the planning of preparations for the next surgery, as it might lead to more precise and accurate estimates. These estimates can be used to make the process more efficiently by notifying OR staff earlier and automatically. The increased efficiency would result in more patients being treated with the same healthcare resources and budget, which reduces the preoperative waiting time.[37 38] The accuracy of the estimates can be improved by extending the LC video data with patient and surgeon specific information from the Electronic Health Record (EHR).[38] An different approach is described by Padoy. with the combination of external cameras and LC videos. This approach extracts more information from one procedure about the positions and movements of the surgeons and OR staff. The additional information is intended to improve the surgical phase and instruments recognition. Still, it is difficult to capture all the members and their movements. The added value of external cameras has not been proven for either patient outcome or surgical efficiency.[37]

1.4 Research questions and aim

1.4.1 Clinical problem definition

In current clinical practice, the preoperative predicted surgery duration is still based on average durations and rough estimates. Due to the large variability in duration of surgical procedures, this results in suboptimal OR planning. On one hand, unexpected longer procedure times induce a delay or even cancellation of subsequent procedures. As a result, patients experience longer waiting times and OR personnel has to work overtime. The increased preoperative waiting time leads to patient discomfort and might even pose a higher risk for complications. On the other hand, unexpected shorter procedure times cause unnecessary vacancy and underutilisation of expensive recourses of the OR. In order to adjust for the large variability in surgery duration, the OR schedulers monitor the duration of each OR either by observation or verbal communication with the OR teams. The OR schedulers estimate the remaining procedure time case-by-case to adapt the OR schedule accordingly. The accuracy of this workflow is highly dependent on the extensive experience with various procedures and the estimation of the OR teams. The process of OR scheduling requires a more robust approach is unbiased, accurate and adaptive.

1.4.2 Research aim

This studies first aim is to predict the remaining procedure duration of LCs by classifying phases of intraoperative laparoscopic videos using a DL network. The laparoscopic images are classified in one of the defined surgical phases of the LC procedure. The phase classifications are used to detect the phase durations. The phase durations are introduced into a ML network to predict the remaining procedure time after each phase. In order to improve OR planning. The second aim of this study is to investigate the importance of adequate labelling for detecting surgical phases of the LC procedure. The performance of a network is affected by both the network structure and the data. Most studies focus only on the development of their networks, rather than analysing their data.

1.4.3 Research questions

1. To what extent is it possible to classify the surgical phases of laparoscopic cholecystectomy procedures in videos using a base-line deep learning network?

2. To what extent is it possible to predict the remaining procedure time of laparoscopic cholecystectomy procedures based on the phase durations using a machine learning model?

3. What is the importance of adequate labelling in phase detection of laparoscopic cholecystectomy procedures?

Primary objective: The development of a data processing pipeline, performance evaluation of a DL network that can classify the surgical phases of LC procedures and ML model that can predict the remaining procedure time based on the phase durations. In an endeavour to improve OR planning.

Secondary objective: Indicate the importance of adequate labelling in surgical phase detection of LC procedures.

Hypothesis: A DL network dedicated for the analysis of intraoperative laparoscopic video data of the LC procedure, will have sufficient accuracy in surgical phase classification to detect the phase transitions. It is expected that adequate labelling of phases in the LC procedure, significantly improves the performance of the classifications made by the network over inadequate labels. The extracted phase durations for the video data, will provide the sufficient information to make predictions about the remaining procedure time. The model will be able to give updates after each phase. The difference between predicted and actual remaining procedure duration is anticipated to be within the set range of five minutes. The model predictions are expected to be closer to the true remaining procedure time than the preoperative estimate, used in clinical practice.

1.5 Study outline

Three investigative steps are essential in order to develop an AI network that predicts the LC remaining procedure time and updates the estimate during the procedure based on the progress. The first element of the study consists of creating an adequate LC dataset. For the LC dataset, the previous mentioned Cholec80 dataset is used. The importance of adequate labelling is assessed by comparing the network performance on the original annotations and annotations according to a revised annotation guide. The second element is the selection of an appropriate phase detection DL network as baseline with suited hyperparameters and desired output format. The output must visualise the network performance, phase classifications and the phase transitions. The phase durations can be obtained by detecting the phase transitions in the LC procedure. The third and last element is the selection of a ML network for the prediction of the remaining procedure time. The network uses the duration of the phases as input. The remaining procedure time is predicted after each phase has past and the phase duration is obtained. The model uses the phase duration of all the past phases for the prediction of the remaining procedure time.

CHAPTER 2

2. Technical Background

This chapter provides a brief introduction into the DL network structures used in the first part of the study for phase recognition, a convolutional neural network. The network classifies the video data of the LC procedure in the surgical phase. In addition, the selected hyperparameters and the network optimisation techniques for this study are explained. The next section describes the ML network used in the second part of the study for predicting timeseries, regression models.

2.1 Convolutional neural network

A convolutional neural network (CNN) is a deep learning network based on the working of the neurons in the visual cortex. This specific network type is most suitable for analysing images. The four basic elements of a CNN are convolutional layers, an activation operation, pooling layers and fully connected layers, shown in figure 2.1. All elements are discussed in detail below.

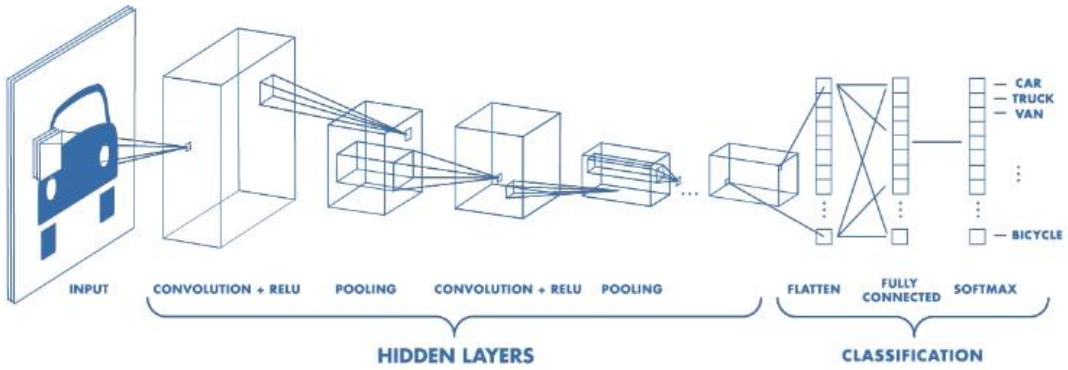

Figure 2.1: Basic network structure of a convolutional neural network.[39]

2.1.1. Convolutional layers

A convolutional layer consists of multiple neurons. Rectangular groups of neurons, with a pre-defined sizes, operate as a filters (kernels) for the pixel values of an input image. The input image is resized to match the optimal dimensions to be fed to the convolutional layer. The number of neurons (nodes) in the convolutional layer determines the width and the amount of layers "the depth" of the network. When a kernel with size 5x5 moves over the input image with step size (stride) one, the dimensions of the output (feature map) are downsized by four pixels.[39] The feature map consists of values that correspond with the degree of similarity that was detected in the input image. Each convolutional layer of a CNN consists of a

lot of kernels, as can be seen on the example presented in figure 2.2. The size and number of kernels can change between the convolutional layers. The kerels of "shallower" layers are detecting mostly lines and "deeper" layers large conceptual structures. The number of kernels and the amount of convolutional layers of a CNN determine the number of different properties that can be detected in each input image.[39]

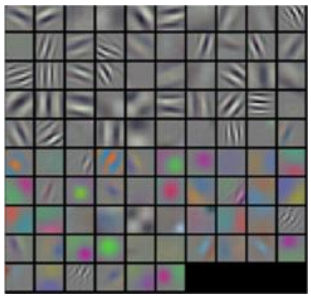

Figure 2.2: Visualisation of possible kernels.[39]

2.1.2 Activation function

The activation function of a neuron is needed in order to process the input (feature) information, as outlined in figure 2.3. The activation functions are non-linear in order for neural networks to approximate complex functions. The inputs (X) and a bias are multiplied by weights (W). The bias shifts the activation function by adding a constant to the input. The bias prevents that the network will only train over point passing through the origin, which has limited flexibility in searching through the solution space. The bias is not connected to the previous layers in the CNN. The inputs and bias are summed before being parsed through to the activation function. When the summation is higher than the threshold of the activation function, the neuron will be activated.

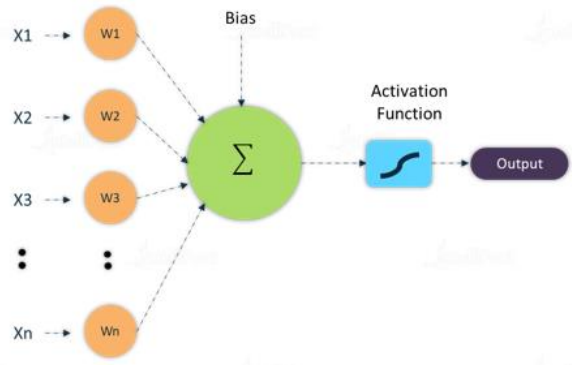

Figure 2.3: Visualisation of a neuron in a CNN.[40]

The two most used activation functions for classification problems are the rectified linear activation unit (ReLU) function and the sigmoid function, shown in figure 2.4. The ReLU function is mostly used for the convolutional layers and the sigmoid function is more often used in the last layer of a neural network. The ReLU function combines the simplicity of a linear activation function and prevents that weighted inputs with negative values can activate the neuron, see figure 2.4. It is important that a neuron will not be activated when the inputs will not contribute to the classification of a class.[41] The ReLU function will eventually result in the network converging towards zero, an optimum in the learning process. The benefits of the ReLU function are that it is simple, either zero or a positive value. There are no additional computations

needed, which speeds up the training process.[42] The sigmoid function is a logistic function that produces an outcome value between zero and one. This makes the sigmoid function suited to create the probability for binary classifications of a network output in the last layer. A threshold of 0.5 is often used to determine which class is assigned to the input image. In contradiction to the ReLU function, the sigmoid function can be activated by negative weight values, see figure 2.4. The last fully connected layer of a network for a multiclass problem is often a softmax layer, with the same number of neurons as classes. The outcomes for a multiclass problem are classified with a value between zero and one. The difference with the normal sigmoid function is that the sum of all the classifications by the neurons is one. The class of the neuron with the highest value is assigned to the input image.[42]

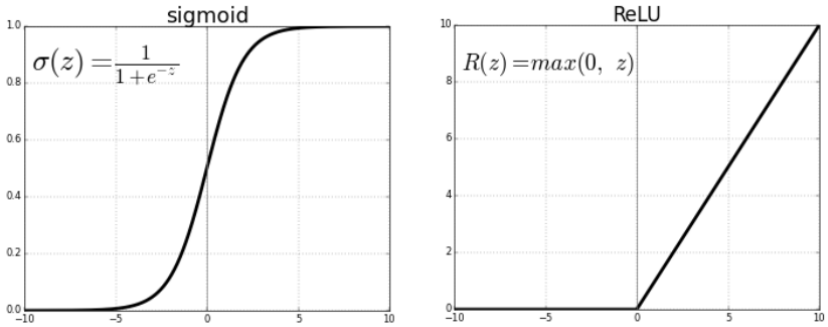

Figure 2.4: Sigmoid and ReLU activation functions with the input weight values on the x-axis and output values on the y-axis.[42]

2.1.3 Pooling layers

The pooling layer reduces the spatial size of the output from the convolutional layers, the feature map. As outlined in figure 2.5, two types of pooling layers exist which use either max-pooling or average-pooling. Often a kernel size of 2x2 is used, which moves with a stride of two over the feature map. In max-pooling, the pixel with the highest value in the kernel is taken. This results in an enhancement of the brighter pixels. In average-pooling, the average of the four pixels in the kernel is taken. As a result, the brighter pixels are smoother. The feature map is reduced to a quarter of the original size with a kernel of 2x2. This decreases the needed computational power and thereby speeds up computation time to process the data. Pooling also improves the learning process of a network by changing the spatial hierarchies of the features. The window is increased, so that it covers a larger fraction of the input image with a lower resolution.[43] Changing the spatial hierarchies of the features prevents overfitting by creating kernels that are more fitted for context recognition than the recognition of specific detailed features.

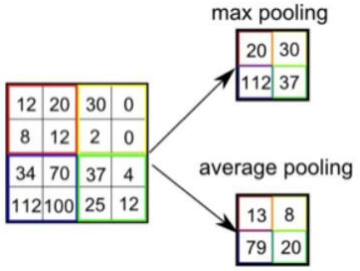

Figure 2.5: Max and average pooling of a feature map.[43]

2.1.4 Flatten layers

The input data of a CNN can consist of colour images, which are three-dimensional. Each pixel in the image has three colour channels being red, green and blue (RGB). The flatten layer is used to process the three-dimensional RGB feature map, created by the convolutional layers of the input images, into a one dimensional feature map. When a feature map consists of 4x4 pixels, as shown in figure 2.5, it is 4x4x3. The flatten layer transforms the 4x4x3 feature map in a 1x48 feature map. Flattening of the feature map is needed to be passed through as an input for the fully connected layers, which takes only one-dimensional data.[39]

2.1.5 Fully connected layers

The final layers of a network are the fully connected (FC) layers. In these layers, the obtained feature information in the previous layers is combined. The input for the FC layers are the flattened activation maps of high spatial features. The information is used to make a classification for the input image on the classes, like a car, truck, van, bicycle etc in figure 2.1. When the network classifies that the image is a car, the activation maps that represent high spatial features of four wheels, lights, bumpers, etc will have high values. The FC layer basically looks at the correlation between the high spatial features of the input image and a particular class. The product between the particular class weights and the output of previous layer, gives the probabilities for the classes. The output of a FC layer with six classes could for example be as follows [0, 0.1, 0.1, 0.75, 0, 0.05]. This output represents a 0% probability for class one and five, 10% for class two and three, 75% for class four, and 5% for class six. The input image has the highest probability for class four of the six classes, which is for example a bike.[39 43]

2.1.6 Dropout layers

Additional layers that can be placed between the FC layers to improve training are dropout layers. The dropout layer nullifies a percentage of the output from the neurons of the previous layer to the next layer. The addition of dropout to a network reduces overfitting during training. The neurons that are nullified change each iteration. The weights of those neuros will not be updated that iteration. Each neuron in the layers tends to specialize in the detection of one specific feature during training. By nullifying the contribution of some neurons for one iteration during training, the other neurons have to anticipate and also learn those features. This results in more generalized and less specialized neurons, which prevent overfitting. Without dropout, the first batch of training data in each iteration influences the learning process more than the later batches. The features that are present in later batches are then under trained.[39]

2.2 Hyperparameters

After the network structure is defined, the network settings (hyperparameters) can be chosen. These hyperparameters define variable settings of the network structure and the training process of the network. They consist of the batch size, number of epochs, loss functions, optimisers, the introduction of dropout, K-fold cross validation, data augmentation, over- and under sampling and post processing. These settings influence the general performance, convergence, and robustness of the network. The general performance expresses the correlation between the classifications and the reference values. The network convergence describes the effort/time needed to reach the optimum of the hyperparameter functions. The robustness

indicates the generalizability of the trained network on other datasets. The choice of hyperparameters should be considered carefully, due to high influence on the network performance.[44 45] The hyperparameters are now further discussed in detail.

2.2.1 Batch size and number of epochs

The event of running the entire training dataset through the network is called an epoch. Most datasets are so large that the data cannot be sent through the network in one time. Sending the data through the network one by one would increase the introduction of noise, which complicates stable training. The dataset is therefore divided into smaller subsets (batches), which are sent through the network. For the training of a network, tens to hundreds of epoch iterations are performed. The batch size effects the duration of the network training and the computational load for the processors. An increase in batch size reduces the number of batches in one epoch and training duration. A larger batch size means that the processor has to process more data at once. The batch size is limited by the computational power of the processor, often a graphical processing unit (GPU) in the computer used for training. A decrease in batch size lightens the computational load and increases the training duration. Another effect of a too small batch size is that it could induce overfitting. The kernels of the network are then trained too specifically on small amounts of data. The optimal batch size has to be found, which is as large as possible but within the limits of the GPU. The final criteria for the batch size is that it has to be a power of two. This is necessary in order to meet the memory requirements for the most efficient calculations.[44 45]

2.2.2 Loss function

Cross-Entropy loss function

In training, a network learns to map the input image to a set of output classes. In this learning process, the search for the optimal network weights is approached as an optimisation problem. The error between the classification and true class is minimised by optimising an error function, which is called the loss function. The loss function must distil all aspects of the network into a single number, in a way that improvement of the number correlates with improved network performance. The loss function is a maximum likelihood estimate, which calculates the mean difference between the classified and true class. The loss function is optimised to an outcome of zero or close to zero. The loss function is used to update the weights to find the those for which the classified classes resemble the reference classes the most. For multiclass classification of surgical phase detection, the Cross-Entropy (CE) loss function is the most suited and used. Entropy is a general term used in data science to describe the quantification of the uncertainty in the possible outcome of an event for a random variable. CE is a measure of the difference between two probability distributions for the total entropy, a given random variable or set of events. It calculates the required number of bits to represent or transmit an average event from one distribution to the other distribution.[46-48] Equation 2.1 presents the CE loss function for binary, $n = 2$, or multi-class problems, $n > 2$. With t_i being the ground truth for that class, p_i being the probability for that class generated by the network and n the number of classes.[49]

$$L_{CE} = -\sum_{i=1}^{n} t_i \, log \, (p_i)$$

Equation 2.1: The Cross-entropy loss function for n classes.[49]

Class weighting

Real-world datasets often have an imbalanced distribution of the data over the classes. Some classes have substantially more data than others. Conventional training of networks on an imbalanced dataset will result in overfitting on the class(es) with the majority of data and underfitting on the class(es) with minority of data. An often-used technique in DL for dealing with imbalanced datasets is class weighting. For a multi-class problem, the class weight of each individual class has to be calculated. The class(es) with more data will have a smaller class weight than classes with few data. The class weights are calculated with the formula of equation 2.2. The *number of class samples* is the amount of data with that class in the dataset. The *total number of samples* is the amount of data from all classes in the dataset.[50] [51]

$$Class\ weight = 1 - (number\ of\ class\ samples / total\ number\ of\ samples)$$

Equation 2.2: The class weighting formula for imbalanced datasets.

The class weight is a factor that is inversely proportional to the amount of data within that class. This factor can be used to weigh the loss computed for the samples of each class during training. By weighing the loss of the classes with an inversely proportional factor, relatively higher weight can be assigned to the loss of the samples from minority class(es). The network will train harder on the minority class(es), which reduces the tendency of the network to overfit on the majority class(es). A negative side-effect of class weighting is that it can introduce a bias. When very high class weights are assigned to the minority class(es), chances are that the network will get biased towards the minority class(es). This will increase the errors in the majority class(es). The performance of the majority class(es) should therefore be monitored. However, the disadvantages of applying class weighting are less than normal training on an imbalanced dataset. Equation 2.1 and 2.2 are combined in equation 2.3 for the function of class weighted CE loss. The L_{CE} loss of each individual class is multiplied by the class weight of that class.

$$L_{CW\ CE} = Class\ weight\ (-\sum_{i=1}^{n} t_i \, log \, (p_i))$$

Equation 2.3: The class weighted Cross-Entropy loss function.[51]

Over-sampling

The previously described technique of class weighting is often used to handle imbalanced datasets. Another technique to deal with imbalance datasets is called resampling. Two resampling methods exist, namely under- and over-sampling. Both methods are presented in figure 2.6. Under-sampling removes samples from the class(es) with the majority of images to match the minority class(es). Over-sampling adds

more similar samples of the class(es) with the minority of images. Resampling balances the number of images from each phase that are used to train the network. Resampling should only be applied on the train set. These resampling techniques also have their disadvantages. Over-sampling could result in overfitting on the minority class(es), since the added data samples are often generated from previous ones and therefore reduce the variance in the class(es). This might introduce a bias towards the minority class(es). For under-sampling, random images from the class(es) with the majority of images are removed. This, in turn, results in loss of information.[52] The most used implementation of over-sampling is by duplicating random images from the class(es) with the minority images but also higher sampling rates of the source data could be used. The network will improve on the classification of the class(es) with the minority of images and should maintain the performance on the class(es) with the majority of images. The performance of the majority class(es) should therefore be monitored. However, the disadvantages of applying resampling are less than normal training on an imbalanced dataset.

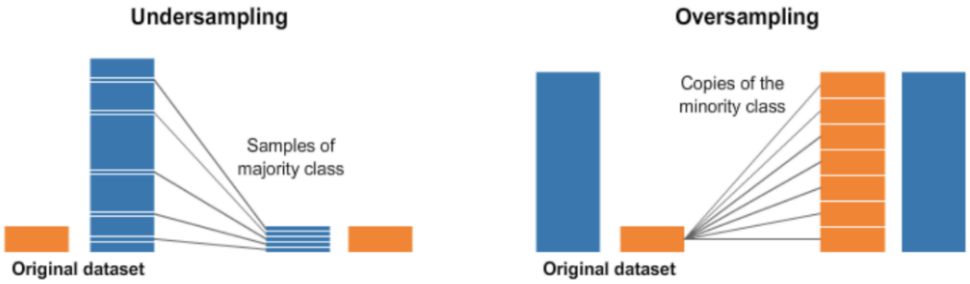

Figure 2.6: Under- and over-sampling to balance the dataset.[52]

2.2.3 Optimisers

Gradient Descent

The improvement of the network results during training is performed by minimising the loss function, which resembles the error between the classified output and the reference. For the minimisation process of the loss function a gradient descent optimization algorithm is used. The error is calculated after each batch during training, which is used to update the kernel weights. The weights are adjusted based on their contribution to the error. This process of weight updating is called backpropagation. Partial derivatives are used to calculate the contribution of the kernels in the last layer to the error. The outcome for the contribution of this layer is used to calculate the contribution of the previous layer and so on. The weight updating process searches for the optimal value that minimises the error. The following types of gradient decent use different approaches, which differ in the moment when the weights are updated during training. Batch gradient descent updates the weights after each epoch. Stochastic gradient descent (SGD) updates the weights after each individual sample of the training set, so one-by-one. The last is called mini-batch gradient descent. The weights are updated after each batch in an epoch. This combines the computational efficiency from batch gradient descent with the speed of SGD, leading to a more precise network and improves the results. Mini-batch gradient descent is therefore the most used optimiser when training with large datasets. However, batch gradient descent does requires more memory for saving the results after each batch. Which reduces the training speed of the network.[53]

Momentum

Momentum is a factor that can be applied to the gradient descent vector, which moves it towards the minimum and reduces oscillations as shown in figure 2.7. The vector is updated with the recent gradients, which are most important. The momentum accelerates when the updates are in the same direction towards the minimum. Combining the current vector with previous vectors reduces the oscillations of the gradient, since the used gradient vector is averaged. The step size towards the minimum is enlarged which causes the gradient to move faster to the minimum.[53]

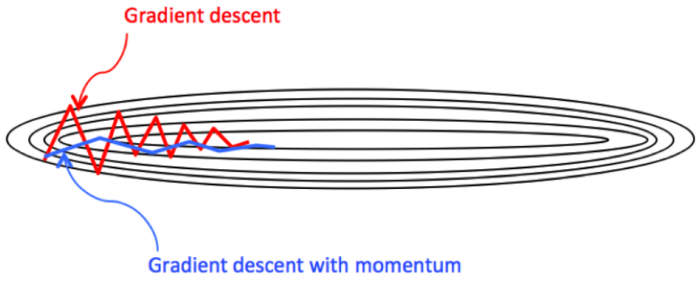

Figure 2.7: The effect of momentum on gradient descent.[53]

Learning rate

The learning rate determines the amount at which the current weight values of a kernel are adjusted, based on the changes in the loss. The learning rate must be chosen wisely as it effects the learning abilities of the network. When the learning rate is large, the weights will converge rapidly but are not able to reach the minimum in the loss, as shown in figure 2.8. This results in an unstable learning process and suboptimal weights. On the contrary, with a really small learning rate the weights will converge slowly. The loss can also have multiple local minima and one global minimum. Either an to large or to small learning rate might result that the network is not able to reach the global minimum but gets stuck at an local minimum. Selecting the optimal learning rate is an important factor in reaching the optimal network weights. Figure 2.8 illustrates the gradient descent of the loss with one global minimum at a small and large learning rate.[53]

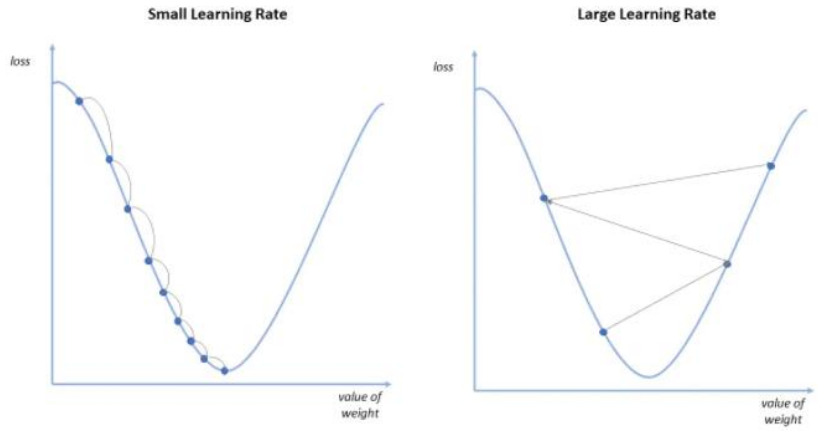

Figure 2.8: The influence of the learning rate on the ability to find the global minimum of the loss function.[53]

Learning rate decay

Learning rate decay uses an adjustable learning rate. In the beginning of the training the learning rate is large and reduces over time during the training. This technique combines the positive aspect of fast converging from a large learning rate and the accurate updating to find the optimal weights from a small learning rate. The large learning rate in the beginning reduces the training time of the network. The small learning rate at the end prevents that the network will overshoot and not find the minimum of the loss with the optimal weights for the network. The decay of the learning rate either follows a predetermined schedule that applies a lower learning rate every epoch, batch of time period or use an exponential decay which uses an exponential function that reduces the learning rate over time.[53]

Adaptive Moment Estimation

Currently, the most used optimiser in DL is the Adaptive Moment Estimation (Adam). Adam was proposed in a paper from Kingma et al. from the university of Amsterdam.[54] Adam combines the Root Mean Squared propagation (RMSprop) and momentum optimiser. The RMSprop uses an adaptive learning rate by applying a moving average of squared gradients in order to normalize the gradient. The normalisation of the gradient prevents an exploding gradient for increasing gradients and vanishing gradient for decreasing gradients. RMSprop uses learning rate decay by reducing a parameter over time during training. Momentum optimiser is added for acceleration of the weight updating towards the minimum. Equation 2.4 presents the function of Adam for updating the weights. θ_n is the initial or previous weight value. θ_{n+1} is the update weight value. α is the step size. υ is the learning rate with decay over time. ε is a constant that prevents that α divided by zero when υ becomes zero. m_n is the added momentum.[55]

$$\theta_{n+1} = \theta_n - \frac{\alpha}{\sqrt{v_n + \epsilon}} m_n$$

Equation 2.4: The function of Adam optimiser for updating the weights.[55]

In this section the elements of a CNN, the hyperparameter settings that can be applied during training and the optimisers that can be used to find the optimal network weights were explained. The next section describes three regression models that are used to predicting timeseries.

2.3 Regression models

Thee regression models are used to predict timeseries for the remaining procedure time of the LC procedure based on the duration of the surgical phases.

2.3.1 Simple and multiple linear regression model

A linear regression model is a ML network. In general, the model assumes that there is a linear relationship between the input variables (*x*) and the single output variable (*y*). The value of y can therefore be calculated from a linear combination of x. Hence, linear regression is a linear approach in modelling the relationship between a dependent variable (*y*) and one or more independent variables (*x*). The case of one independent input variable is called a simple linear regression (SLR), given by function 1 of equation 2.3. The variable *w* is the weight factor that determines the slope, *b* is the intercept with the y-axis at *x* is zero,

and ε the random error. When there are multiple independent input variables, it is referred to as multiple linear regression (MLR). The MLR is given by the function 2 of equation 2.3, w_1, w_2, etc are the weight factors for each x that determine the slopes, b is the intercept with the y-axis at x is zero, and ε the random error.[56] [57]

$$y = wx + b + \varepsilon \quad (1)$$
$$y = w_1x_1 + w_2x_2 + \cdots + b + \varepsilon \quad (2)$$

Equation 2.3: The SLR and MLR model functions.[57]

2.3.2 Random Forest Regression model

A Random Forest (RF) is an ML technique which can be used for both regression and classification tasks. A RF consists of multiple Decision Trees (DT), that can either output a categorical or numerical prediction. The DTs of a RF are also known as Classification And Regression Trees (CART). The building blocks of a DT are nodes and branches. The nodes serve as evaluation points were one of the features in the data are evaluated by a threshold, when making a prediction. The DTs of a RF are sensitive to the specific data they are trained on. Bagging is a ML data sub-sampling technique involving replacement. The DTs are trained on the sub-samples, creating variance in the output.[58] The training process consist of searching for the node with the threshold that splits the data in the best way. Categorical trees use entropy as evaluation metric, regression trees use the Mean Squared Error (MSE). The evaluation is different for discrete and continuous features. For discrete features, all possible values are evaluated for each variable by the metric. For continuous features, the average of each two consecutive values in the training data are applied as possible thresholds. There are three types of nodes being root, intermediate and leaf nodes. The root node is the first node of the tree and evaluates which variable splits the data in the best way. The intermediate nodes also evaluate variables but do not make predictions. The leaf nodes are the last nodes of the tree, that make the predictions of a category or numerical value. There are two hyperparameters that specify the training process of a RF, being the maximum depth and number of estimators. The max. depth refers to the max. number of consecutive nodes of each DT and the number of estimators are the max. number of DTs in the RF. After the DT is trained, the categorical or numerical value can be predicted for of a new sample. The DT starts at the root node and based on the value of the feature that is evaluated, go to the left or right to the intermediate node. The same process is repeated for the other intermediate nodes until a leaf node is reached. Depending on the type of problem two things can happen. For a classification tree, the predicted category has the highest probability of the categories that are on the leaf node. In the case of a regression tree, the prediction is the average of the values for the target variable on that leaf node.[59] Figure 2.9 shows a RF where the instance represents the bagged data that is introduced at the root node. The figure shows the different paths through the DTs, resulting class classifications. The final class will be determined by majority-voting.

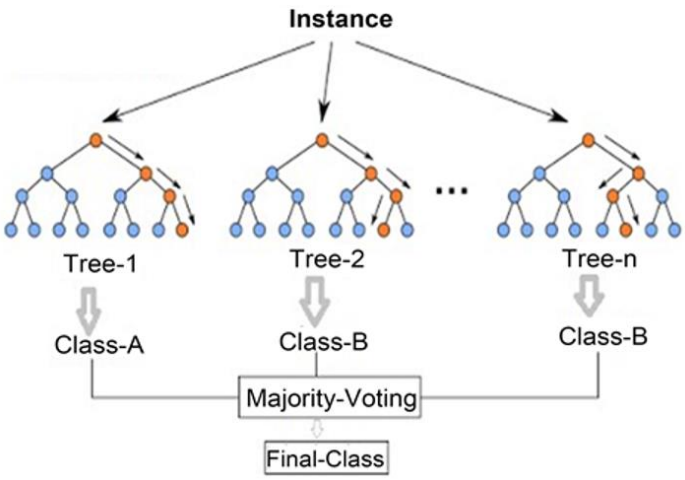

Figure 2.9: Categorical RF regression model with n-number of decision trees.[59]

2.3.3 Support Vector Regression model

Support Vector Machines (SVM) are often used and well known in the ML community for solving classification problems. The application of SVM for regression problems are known as Support Vector Regression (SVR) models. SVR applies one of the following functions to solve the regression problem; linear, polynomial, radial basis function or sigmoid, illustrated with an example in figure 2.10. The polynomial function is a higher order, to the power of ≥ 3, function. The radial basis function (RBF) is a function that bases the output value depending only on the Euclidean distance between the input and a reference point, being either the origin or some other fixed point. The objective of SVR is, as in most regression models, to minimize the sum of squared errors. SVR uses Lasso (L1), Ridge (L2) or ElasticNet, which are all extensions on least squares error that include an additional penalty parameter aiming to minimize the coefficients. The error term is handled within the constraints, the specified error margin (ϵ). The constraints can be tuned in order to gain the desired accuracy.[60] [61]

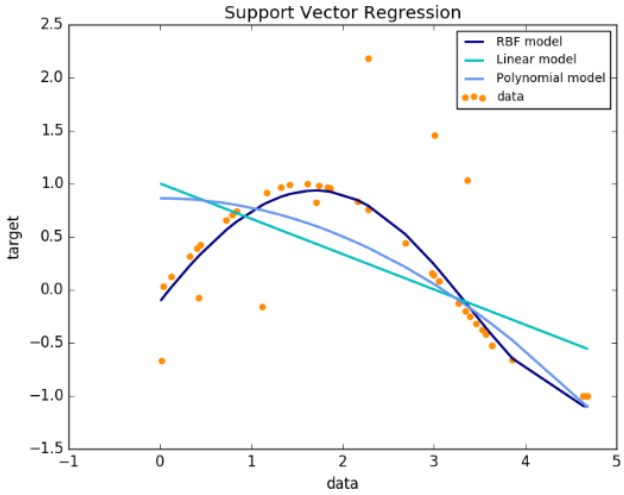

Figure 2.10: SVR model predictions with variating functions on example data.[60]

CHAPTER 3

3. Surgical phase detection of laparoscopic cholecystectomy procedures

This chapter describes the first part of this study, which is the development a DL network that can accurately and objectively classifying the surgical phases of intraoperative LC videos. The classification results can be improved by post processing. In addition, the importance of adequate labelling of surgical video data are shown by comparing two annotations of the same LC video data. After classification, the duration of the individual phases can be extracted by detecting the phase transitions. The duration of the individual phases serves as an input for the second part of this study.

3.1 Introduction

3.1.1 Surgical phase detection

The interest in automated surgical phase detection for minimally invasive surgery, such as LC, has increased in the recent years. It has become a cornerstone for the realisation of AI applications in surgery. A common definition for surgical phases is, higher-level tasks of the surgical procedure, e.g. the dissection of Calots' triangle in order to achieve the CVS in the LC procedure. Surgical phase detection is used for workflow recognition to improve the learning curve of residents or the performance of surgeons. Another application is surgical process modelling, which provides the possibility to automatically gather the available information in the surgical procedure. That information can lead to potential improvements in OR logistics and surgical patient care. The automatic identification of specific actions, procedure steps, or adverse events can be used to make predictions about procedure duration or chance of complications. In this study, the surgical phase detection will be used to automatically detect the phase transition which can be related to the duration of the individual phases of the procedure. This information can later be used to make predictions that could improve the OR logistics.

3.2 Technical background

3.2.1 Neural networks

There are multiple types of neural networks such as a Convolutional Neural Network (CNN), Temporal Convolutional Network (TCN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) and Hidden Markov Models (HMM). A CNN has been developed for automatic and adaptive learning of the spatial hierarchies of abstract features in images for efficient object identification through backpropagation.[62] A TCN performs causal convolutions, no "leakage" of information from future to past, and dilations in order to adapt for the temporality of the sequential data.[63] A LSTM is a type of RNN that is capable of learning order dependence in sequence classifications and store information for a longer period of time.[64] A HMM is a statistical model that describes two stochastic processes, being, the evolution of

observable events called ‘symbols’ that depend on invisible internal factors called ‘hidden states’. The hidden states, forming a Markov chain, depend on the probability distribution of the observed symbols.[65]

3.2.2 Network performance

The performance evaluation of the network can be assessed by comparing the manually annotated phases, ground truth, with the automatically recognized phases in the validation dataset. The results from this comparison are presented in a confusion matrix (CM) with the true positives (TP), true negative (TN), false positives (FP) and false negatives (FN) recognitions. A CM is also capable of showing the performance relations in multi-class classifications. The CM shows the amount of each class that is classified correctly, but also the amount that is falsely assigned to the other classes. An example of a CM for a binary and multi-class classification with fabricated data is presented in figure 3.1.[29] In order to limit the bias in the performance assessment, the ground truth and network classifications are compared on a test dataset. The test dataset is not used for the learning process of the network and is thus unknown. The latter is of great importance because the network could be trained too extensively on the training and validation dataset, which can result in overfitting. When new data are introduced to the network, the performance will be lower than on the validation dataset.

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Class	1	2	3	4	Total
1	80	10	8	2	100
2	0	75	11	14	100
3	3	20	77	0	100
4	5	7	4	84	100

Figure 3.1: Two example confusion matrices are presented. The left matrix shows the relation of the network performance for a binary classification. The right matrix shows the relation of the network performance for a multi-class, in this case four, classification with fabricated data. The matrix gives for each class the amount of correctly classified frames and the amount of frames that are classified falsely to other classes.

Commonly used performance metrics in DL are the accuracy, precision, recall and F1-score. The percentage of frames that the network recognises correctly is given by the accuracy, calculated with function 1 of equation 3.1. In other words, the accuracy is the probability that the network classifies the correct class for a randomly selected unit of the dataset. The condition for this metric is that the data has to be balanced, even amount of negatives and positives, to give the appropriate performance of the network. Precision indicates the percentage of all positively classified frames in which the network recognized a phase correctly, calculated with function 2 of equation 3.1. The percentage of all actually positive frames that are correctly classified as positive by the network is given by the recall, which is the same as the metric sensitivity. The recall is calculated with function 3 of equation 3.1.[21] Finally, the F1-score is the weighted average of the precision and recall, which is also known as the dice-coefficient. It shows the balance between precision (exactness) and recall (completeness) of the network in one metric. The F1-score is especially valuable with imbalanced data sets, as the accuracy might be misleading. The calculation of the F1-score is given by function 4 of equation 3.1.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$
$$Precision = TP / (TP + FP) \quad (2)$$
$$Recall = TP / (TP + FN) \quad (3)$$
$$F1\text{-}score = 2 * Precision * Recall / (Precision + Recall) \quad (4)$$

Equation 3.1: The functions for the accuracy, precision, recall and F1 score.[21]

Stauder et al. visualised the output of their network in a barplot with the classified and ground truth for the surgical phases during the duration of the procedure.[66] Figure 3.3 presents the results of one laparoscopic video of their test dataset.

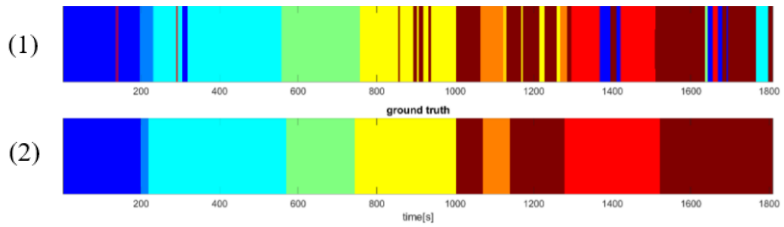

Figure 3.3: The figure shows an example of the colour-coded classification output of a network for eight surgical phases (1). As a reference are the ground truth of the surgical phases presented below (2).[66]

In addition to these performance metrics, class activation maps (CAM) are used to evaluate CNNs. The CAM visualises the regions in the input image that have the highest informative value for the classified class. The CAM is a heat map highlighting the pixels of the input image that trigger the network in associating the image with that specific class. This gives more understanding and enables analysis of the informative regions for possible bias in the data that influences the performance of the network. The technique to produce the CAM relies on global average pooling (GAP) layers, introduced after the final convolutional layer of the CNN. The output of the final convolutional layer are N feature maps. The GAP layer takes the N feature maps as input and returns the spatial average, where higher activations are represented by higher signals. The GAP layers spatially diminish the feature maps of the image and gives a datapoint per feature map. The CAM is a linear combination of average pooled feature maps and is up-sampled in order to match the size of the input image. A possible drawback of the CAM is that it is constrained to the visualization of latter stages in the image classification.[67] As an example for CAM in surgical image recognition, the results of Namazi et al. are shown in figure 3.4.[68] They visualised the CAMs on the output of their recurrent CNN for the surgical tool recognition on LC video data.

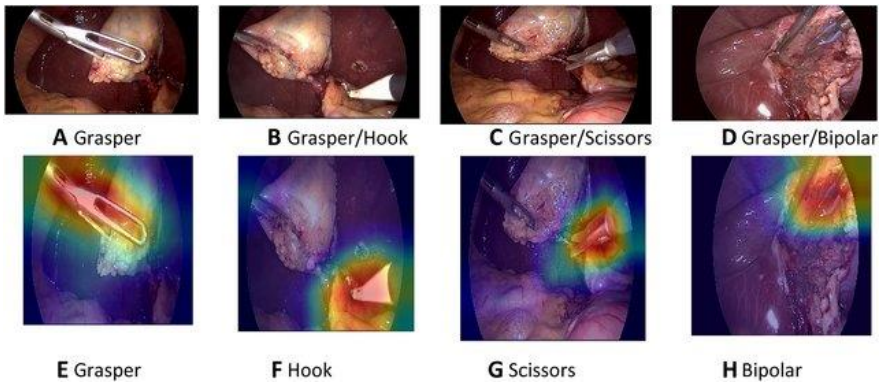

Figure 3.4: The figure shows examples of class activation maps on images of a laparoscopic cholecystectomy for surgical tool recognition. On the top are the original images and on the bottom the images with class activation overlay.[68]

3.2.3 Previous research

Various studies have researched the possibilities of using deep neural networks for automatic phase recognition. Some previous studies worked on retrieving information on the progress of the LC procedure by recognizing seven surgical phases with extracted abstract visual features from the intraoperative laparoscopic videos using CNNs.[21 22 24 30 36] Guédon at al. showed a 79% precision with a CNN.[21] The hybridization of a CNN with a HMM introduces sequence information as an additional factor on the extracted features for the phase classification of the LC procedure. In hybrid networks the output of the first network, based on the input data, severs as an input of the second network. The HMM incorporates the probability whether the current frame should transition from the phase of the previous frame to the next phase. The probability of the transition to the next phase increases as the duration of the phase increases. The most common workflow under surgeons is used for the order of transitions.[69] Twinanda et al. showed a 91% precision and 86% accuracy using a combination of a CNN and HMM.[30] The HMM can also be used to predict the remaining duration of the LC procedure. Other previous studies worked on retrieving information about the progress of the LC procedure by recognizing the seven surgical phases from the intraoperative laparoscopic videos using a TCN or a combination between a CNN and LSTM. The hybridization of a CNN with a LSTM introduces memory in the network that captures the spatial and temporal correlations in the laparoscopic video data, for improved phase classification of the LC procedure. Yengera et al. showed an accuracy of 83% and precision of 78% using a CNN with a LSTM.[70] The use of a TCN for the phase detection has the advantage that the network has a large temporal receptive field. The TCN is able to capture the full temporal resolution with a reduced number of parameters. This allows for faster training and use the temporal information optimally. Czempiel et al. showed with a Multi Stage - TCN an accuracy of 89% and precision of 82%.[71] Hong et al. investigated the annotation generation process for surgical phase recognition on 24 videos of the Cholec80 with a CNN-LSTM and 3D-CNN. The revised annotations showed an improvement between 2 - 5% in average precision compared to the original annotation.[35] From these studies can be concluded that CNNs show decent performance for surgical phase recognition of LC procedures and that the use of temporal information improves the performance even further. In addition, the reannotation of surgical phases in LC data can have a positive effect on the network performance.

3.3 Materials and Methods

3.3.1 Intraoperative dataset

Cholec80

The intraoperative laparoscopic video data to train the DL network in the classification of the phases of the LC procedure is acquired from open source data. The data source is the Cholec80 dataset from the University Hospital of Strasbourg, made publicly available for further research by Twinanda et al. The dataset consists of 80 laparoscopic videos of LC procedures performed by 13 surgeons. The original annotation of the Cholec80 has seven phases, shown in figure 3.5 and in table 3.1 with their accompanying surgical tasks and duration. The dataset is annotated at 25 fps by a senior surgeon.[30]

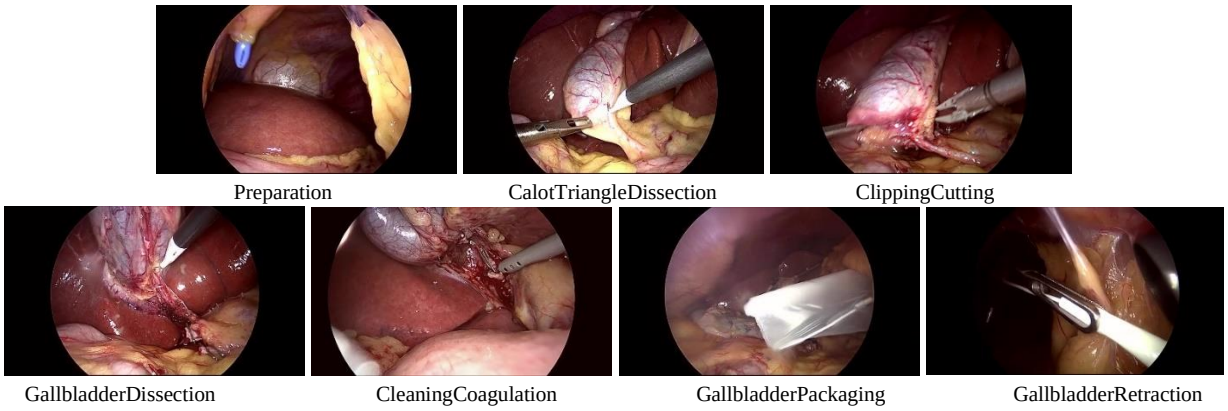

Figure 3.5: The original phase definition of the Cholec80 dataset.

The first phase is Preparation, in which the trocars and instruments are introduced in the AC. The second phase is CalotTriangleDissection, here the bile duct and artery are dissected to gain the CVS. The third phase is ClippingCutting. After gaining the CVS, the common bile duct and artery are clipped and cut. The fourth stage is GallbladderDissection, here the gallbladder is detached from the liver. The fifth phase is CleaningCoagualation. This phase can take place at multiple moments during the procedure. It often takes place after the ClippingCutting and GallbladderDissection to remove leaked bile or stop a bleeding. It can also occur in CalotTriangleDissection, in the case of a bleeding. The sixth phase is GallbladderPackaging, in which the gallbladder is placed in the bag. In the seventh and last phase, GallbladderRetraction, the bag is removed from the AC.

TABLE 3.1

ORIGINAL CHOLEC80 PHASE ANNOTATION WITH THE SURGICAL TASKS

Phase of the LC procedure		Surgical tasks	Duration in seconds
1	**Preparation**	Create pneumoperitoneum with Veress needle Insert trocar for laparoscope Insert laparoscope through the trocar Insert other three trocar under direct sight Insert the graspers through the trocars	125 ± 95
2	**Calot triangle dissection**	Dissect adhesions to the GB Dissect and mobilize Hartmann's pouch Dissect and isolate the CD and CA (CVS)	954 ± 538
3	**Clipping & cutting**	Place two clips on the proximal end of the CD and CA Place a clip on the distal end of the CD and CA Transect the CD and CA between the clips	168 ± 152
4	**Gallbladder dissection**	Dissect medial side up to the fundus of the GB Dissect lateral side up to the fundus of the GB Dissect the under surface of the GB from the liver	857 ± 551
5	**Cleaning & coagulation**	Coagulate any bleeding site Clean any blood or leaked bile Check the clips on the CD and CA stumps	178 ± 166
6	**Gallbladder packaging**	Retract a trocar from the abdominal wall Insert retrieval bag through the incision Place the GB in the back and close the bag	98 ± 53
7	**Gallbladder retraction**	Place the stretcher in the incision with the retrieval bag Stretch the incision Retract the retrieval bag through the incision Retract all trocars from the abdominal wall Deflate the pneumoperitoneum from the abdominal wall	83 ± 56

Table 3.1: Original phase annotation of the LC procedure with the according surgical tasks and duration, mean and std. [30 72]

3.3.2 Data annotation

Flaws in original Cholec80 phase annotation

The Cholec80 has been used in many studies of surgical phase detection in LC procedures. However, there are some flaws in this dataset. The first downside is that no annotation guide is provided for their annotation of the dataset. As described in table 3.1, the surgical tasks of the phases are defined but no information is available about the exact cut-off points of each phase. There are no guidelines when new data are added to the original 80 video and needs to be annotated. The second downside is that the phase annotations are inconsistent in the beginning and ending of certain phases. This means that in some videos a portion of the images are mislabelled. This increases the amount of noise in the dataset substantially, which influences the results. The third downside is that the images outside the AC are also labelled the same phase as the images in the AC. These images are very similar in each phase and are not specific to that phase, as can be seen in figure 3.6. When looking at these images individually, even an experienced surgeon could not identify to which phase they belong. So let alone a neural network is capable of

classifying them correctly. In the Cholec80, there are over 5500 images out of the AC at one fps. This introduces a structural error in the classifications of the network. The fourth and last downside is, that the phase CleaningCoagulation contains two actions. These actions are also part of several other phases and should therefore not be considered as an individual phase. This annotation unnecessarily increases the amount of phase transitions.

Figure 3.6: Two images out of the AC of different surgical phases in the Cholec80 dataset.

Revised definition for phase annotation

For this study, revised phase definitions were composed which was needed due to the previously mentioned flaws in the original annotation and the possibility for data acquisition from the MMC in continuation research. The revised phase definition consists of six phases, selected on clinical relevance and technical capabilities of the network. The phase definitions are defined, annotated and double-checked by a surgeon and technical physician. The phases consist of five surgical phases: Preparation, Exposure and Dissection of Calot's Triangle, Clipping and Transection of cystic duct and artery, Gallbladder Dissection from hepatic plate / fossa, and Hemostasis, Packaging and Retraction of gallbladder. The sixth phase is an additional phase: Out of Body.

The phase definitions were standardized through an annotation guide, presented on the next page. Phases and annotation guide were defined by expert surgeons and AI researchers, taking into account clinical-relevancy and algorithmically considerations. The surgical phases are defined to simulate the common workflow of surgeons, focusing on the action performed in that phase to reach a specific goal. The surgical tools are often used as cues to define the beginning of the phases. The last surgical phase, HemostasisPackagingRetraction, combines the short individual phases at the end of the procedure in the original annotation of the Cholec80. These individual phases have limited added clinical value but impose technical difficulty. The additional, non-surgical, phase is added to improve the network classification performance, as the out of body images are not specific for any surgical phase. In all previous research with LC data where phases were defined, e.g. Twinanda et al. and Hong et al., this has never been included.[30 35] This introduces a standard error in the data, although it might be a few percent. A detailed description of the redefined phases of the Cholec80 is given in table 3.2. The short last three phases of the original annotation are combined to one phase.

TABLE 3.2

REDEFINED SURGICAL PHASE ANNOTATION GUIDE OF LAPAROSCOPIC CHOLECYSTECTOMY

Surgical phase		Starting point	End point	Description
1	**Preparation**	The first insertion of the laparoscope in AC	The moment before the first grasp of GB with the instrument	- Placement of laparoscopic ports and instruments - Adhesiolysis from abdominal wall
2	**Exposure and Dissection of Calot's Triangle**	The first grasp of GB with the instrument	The moment before the first introduction of clip applier, in order to clip CD and/or CA	- Exposure of the gallbladder, including division of potential adhesions to the GB - Opening peritoneum of the GB - Dissection of CD and CA
3	**Clipping and Transection of CD and CA**	The first introduction of the clip applier, in order to clip the CD and CA	The last moment the scissors is in view during the retraction after transaction of the CD & CD	- Clipping CD and CA - Transection of CD and CA - Including eventual dissection during clipping
4	**Gallbladder Dissection from fossa/hepatic plate**	The first moment after the scissors disappears out of view	The last moment gallbladder is connected to the liver, before final release from hepatic plate	- Dissection of GB from liver bed - Including coagulation of liver bed, irrigation and suctioning before GB is released
5	**Hemostasis, Packaging and Retraction of gallbladder**	The first moment GB is completely released from the hepatic plate	The final view of the AC during the retraction of the laparoscope at the end of the procedure	- Extraction of GB with retrieval bag - Removal of trocars - Suctioning, gallstone retrieval (additionally) - Cautery of gallbladder fossa (additionally) - Drain placement (additionally)
6	**Out of Body**	The first moment the intra-abdominal organs are out of view during the retraction of the laparoscope	The first moment the intra-abdominal organs are in view during the insertion of the laparoscope	- Prior to first introduction in the AC - Cleaning of the laparoscope - White balancing - After retraction of laparoscope from the AC at the end of the procedure

Table 3.2: Redefined phase annotation of the LC procedure with the precise begin and end point definition.

3.3.3 Data processing

Video to frames conversion

For this study, the video data of the Cholec80 and new inclusion are both processed in the same manner. The conversion from videos to frames for the image dataset is performed with FFmpeg for high resolution and the VisionWorks Python package from F. Milletari at one fps.[73] The frames were converted to 854 x 480 pixel and saved as a PNG-file.

Division of dataset

The dataset was split in a train, validation and test set with a ratio of 0.5 : 0.1 : 0.4 respectively, according to the distribution of Twinanda et al. and Czempiel et al.[30 71] The images were split per video, meaning that all images of each procedure were either in the train, validation or test set. The images could not reside in multiple datasets. This resulted in 40 train, eight validation and 32 test videos. These split ratios were used for comparability of results.

Data augmentation

The images from the dataset are transformed in order to fit through the network and improve trainability. First, the transform reads the image as a Python Image Library (PIL). The PIL image is then resized to a width and height of 256 pixels. Next, the resized PIL image is cropped with CenterCrop from Torchvision to a width and height of 224 pixels. This removes most of the black edges created by the round camera of the laparoscope, which reduces the amount of data that needs to be processed. The pixels of the cropped image are then transformed to tensors and normalized with TformWrapper from Torchvision. Normalisation converts the pixel value range from 0 - 255 to 0.0 - 1.0, which is easier to process by the network. TformWrapper uses the following formula to normalize the pixel values: Normalized pixel value = (input pixel value – mean pixel value) / standard deviation. The same mean RGB pixel values from the ImageNet dataset were used [0.485, 0.456, 0.406] and their standard deviation (std) [0.229, 0.224, 0.225]. The last transform was applied on the image labels to convert them in integers and then tensors.

Moving window

The phase classification output of a CNN often contains noise, as can be seen in figure 3.3. The CNN only incorporates the features that are present in a single image. Within a phase, there are images that have features that could fit to multiple phases. These images have a high change of being classified to the wrong phase, which introduces noise to the output. The noise is often a small portion of all the classifications of a phase. Hence, it can therefore be filtered, which will improve the final output of the network. The noise can be filtered with a moving window. The moving window takes a subset of the total output, which are indices that represent phases. The window slides with a step size of one along the array with outputs. The window size determines the number of elements in the subset. The classifications of that subset evaluate, whether the classification of each individual image has to be adjusted. A threshold is used to determine how many classifications of the subset have to be of the same phase, to change the classification of that image to that phase. The length of the individual phases variates, therefore the window size and threshold can be selected for each phase. In most cases, shorter phases benefit from a smaller window and longer phases from a larger window. The threshold is often set at 50% of the window size. The moving windows were applied on the network output in ascending numerical order of the phases. The frame at half the width of the window was adjusted by the filter. Pseudo-code has been provided for the filtering process of the phase classifications.

Pseudo-code for classification and filtering of phases

The discrete time index t runs from 0 to the end time T. The complete time interval is denoted by 0:T. A partial time interval from time i to j is denoted by i:j. The video data consist of frames at discrete time stamps t and can be regarded as a set of measurement. A frame acquired at time t is denote by the variable z_t and the collection of frames in a partial interval is denoted by $z_{i:j}$. The true values of the to be estimated phases of the frames are categorical and denoted by w_t for the frame at time t and $w_{i:j}$ for the collection. The phase are discrete states and have discrete probabilities. The phase estimates of the frames are denoted by $\hat{w}_{t|j}$ at time t and $\hat{w}_{t|i:j}$ for the collection. For t we assume that it is the current time, implying that $t + i$ with $i > 0$ is in the future and $t - i$ with $i > 0$ is in the past. The phase classification by the network for the input frames, z_t, are instantaneous (non-temporal) classifications denoted by $\hat{w}_{t|t}$. The network classifications are post processed by filtering with a moving window that has a fixed lag of k frames using $\hat{w}_{t|t}$ from the l most recent frames as the input. The phase estimates are based on retrodiction making it a causal system, as only classifications of current frames and from the past are used. The processed estimates by the moving window are denoted by $\hat{w}_{t-k+1|t-l+1}$.

3.3.4 Phase detector network design

The chosen base-line network design for the phase detector (PD) derives from the original architecture suggested by He et al in their Computer Vision and Pattern Recognition (CVRP) paper for the ImageNet challenge of 2015.[74] The network is a residual neural network with 50 layers (ResNet50), which incorporates skip connections to jump over one or more layers. Czempiel et al. also used the ResNet50 as a reference network.[71] The ResNet50 consists of one individual convolutional layer, four convolutional blocks, max and average pool layer, and softmax layer, as can be seen in table 3.3. The layers are activated with the ReLU function. Four fc-layers were added between the average pool and softmax layer to gradually reduce the features. Between each fc-layer, three batch-normalisation and drop-out layers were added to improve the trainability and reduce overfitting.

TABLE 3.3
RESNET50 NETWORK ARCHITECTURE

Layer name	Output size	Network layers	Presence
Conv1	112x112	7x7, 64, stride 2	x1
Conv2_x	56x56	3x3, max pool, stride 2	x1
		1x1, 64 3x3, 64 1x1, 256	x3
Conv3_x	28x28	1x1,128 3x3, 128 1x1, 512	x4
Conv4_x	14x14	1x1, 256 3x3, 256 1x1, 1024	x6
Conv5_x	7x7	1x1, 512 3x3, 512 1x1, 2048	x3
	1x1	Average pool, 1000-d fc, softmax	x1

Table 3.3: ResNet50 network architecture with layer blocks with variating amount of convolutional layers.[74]

3.3.5 Network implementation and training

The PD - network was designed and trained using PyTorch (v1.8) and PyTorch-Ignite (v0.4.6) libraries on a Tesla P100-PCIE-16GB, Titan-X 12GB and NVIDIA GeForce GTX 1080Ti. Pretrained ImageNet weights were used to reduce the required number of epochs for training to make the network converge. Hyperparameter optimisation was performed for the batch size, Adam and SGD optimiser and the learning rate with the sweep function of Weights and Biases (v0.12.0) (wandb). For the Original phases, the batch size was set to 68 and the Adam optimiser with a learning rate of $5.447 * 10^{-5}$ was used. The Revised phases were trained with the batch size of 70 and the Adam optimiser with a learning rate of $2.487 * 10^{-5}$. Class weighting was applied on the training set to adjust for the imbalanced distribution of the frames over the phases, instead of over-sampling. Over-sampling affects the loss by alternating the data that is introduced in the network. Class weighting affects the loss in a more even way compared to over-sampling, which was preferred. The class weights for the Original phases are: Preparation 3.28, CalotTriangleDissection 0.33, ClippingCutting 1.68, GallbladderDissection 0.51, GallbladderPackaging 3.32, CleaningCoagulation 1.71 and GallbladderRetraction 3.76. For the Revised phases, the class weights are: Preparation 7.21, ExposureDissectionCalotTriangle 0.37, ClippingTransection 2.29, GallbladderDissection 0.62, HemostasisPackagingRetraction 1.04 and OutofBody 5.61. The one-dimensional batchnormalisation layers were set at 1024, 512 and 256 features, and the dropout at 0.2. The PD was trained for 100 epochs and Checkpoint from PyTorch-Ignite was used after each epoch to save the network weights with the highest validation accuracy. The training, validation and test results were logged were logged to wandb.

3.3.6 Network performance evaluation

The network performance was evaluated during training on the validation set and after training on the test set. The results of the network classifications were presented in a CM for comparison of the performance on the individual phases. It also gives insight into which phase the frames are misclassified. The performance of the multi-class problem was evaluated based on overall and segmental metrics. The overall metrics are the combined results of all phases, and the segmental metrics are calculated of each phase individually. The dataset is imbalanced which could result unrepresentative overall performance results when the network overfits on the majority phases. The inclusion of segmental metrics gives insight into the performance on the individual phases, whether the network is under or over performing. The used performance metrics are accuracy, precision, recall and F1-score. The percentage of frames that the network recognises correctly is given by the accuracy. The precision refers to the percentage of all positively classified frames in which the network recognized a phase correctly. The percentage of all actually positive frames that are correctly classified as positive by the network is given by the recall. The F1-score gives the balance between the precision and recall in one metric. The accuracy might be misleading as the data set is imbalanced. The F1-score is a more suited performance metric in that case. In addition to the performance metrics, CAMs are made of a selection of the frames from each phase. The CAM gives information on the focus of the network in the frames and whether there is a bias in the data that influences the performance of the network.

3.3.7 Phase detection pipeline

A pipeline has been developed for this study, shown in figure 3.7 and 3.8, to process video data of the Cholec80 with the original and revised phase annotation of the LC procedure separately. The video data are converted to frames with a sample rate of one fps. For each data annotation, a separate ResNet50 network is trained. The frames and related phase annotations of the test set are passed through the network after training for performance evaluation. The output of the network are phase classifications for each individual frame from a video of a procedure. The phases are colour-coded and visualised in a barplot with the frame numbers on the x-axis. The performance of the network is expressed in overall and segmental metrics for comparison between both annotations and the results of previous research. The barplot from the direct output of the network is filtered with a moving window. The network also makes CAMs from individual frames of the video, adding a heatmap overlay. These provide information about the location(s) in the frame that the network correlates with the classified phase and show possible biases in the data.

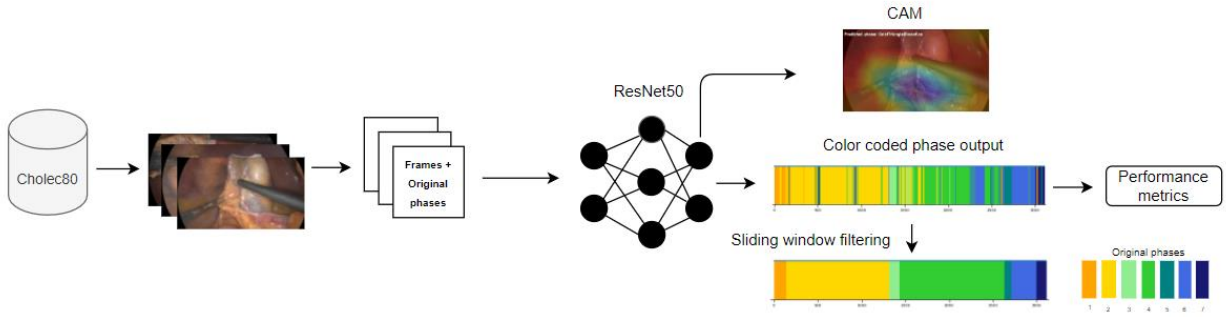

Figure 3.7: Phase detection pipeline with the original phase input, colour coded phase output and CAM images.

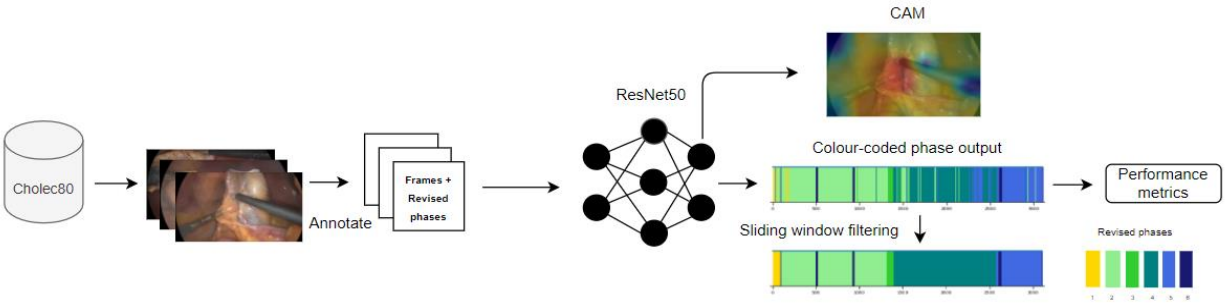

Figure 3.8: Phase detection pipeline with the revised phase input, colour coded phase output and CAM images.

3.4 Results

3.4.1 Data preparation

Original annotation

The original annotation of the Cholec80 dataset defines seven phases for the LC procedure. The video data are sampled at one fps for this study, resulting in a dataset of 184579 frames. The number of frames in each phase: Preparation 5455, CalotTriangleDissection 75201, ClippingCutting 12789, GallbladderDissection 59196, GallbladderPackaging 16567, CleaningCoagulation 8378 and GallbladderRetraction 6992. The distribution of the frames over the phases is also visualised in figure 3.9.

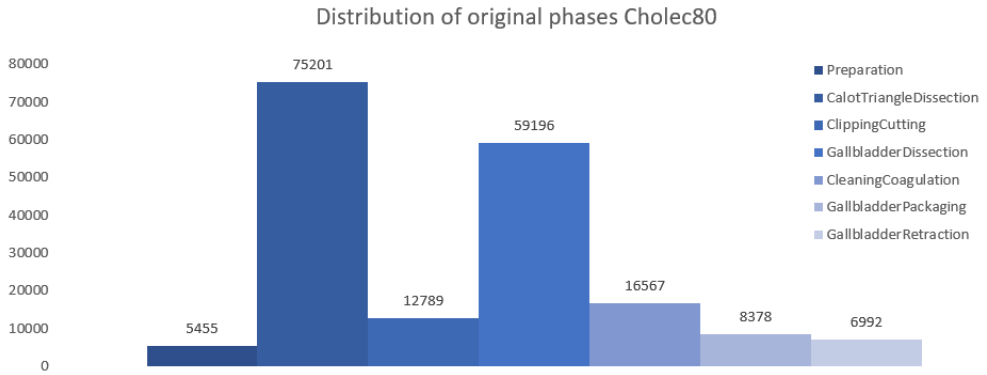

Figure 3.9: Distribution of the frames over the phases of original Cholec80 annotation at one fps.

Revised annotation

The revised annotation of the Cholec80 dataset defines five surgical phases and one non-surgical phase for the LC procedure. The revised phase definitions are visualised in figure 3.10.

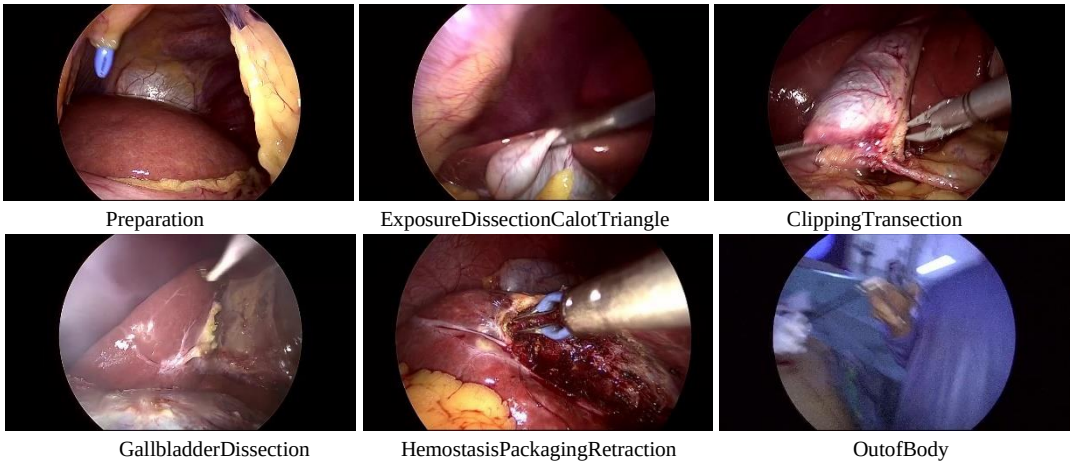

Figure 3.10: The visualisation of the revised phase definitions for the Cholec80 dataset.

The revised annotations of the Cholec80 video data are sampled at one fps for this study, resulting in a dataset of 184579 frames. The number of frames in each phase: Preparation 3703, ExposureDissectionCalotTriangle 78281, ClippingTransection 13096, GallbladderDissection 56057,

HemostasisPackagingRetraction 27915 and OutofBody 5526. The distribution of the frames over the phases is also visualised in figure 3.11.

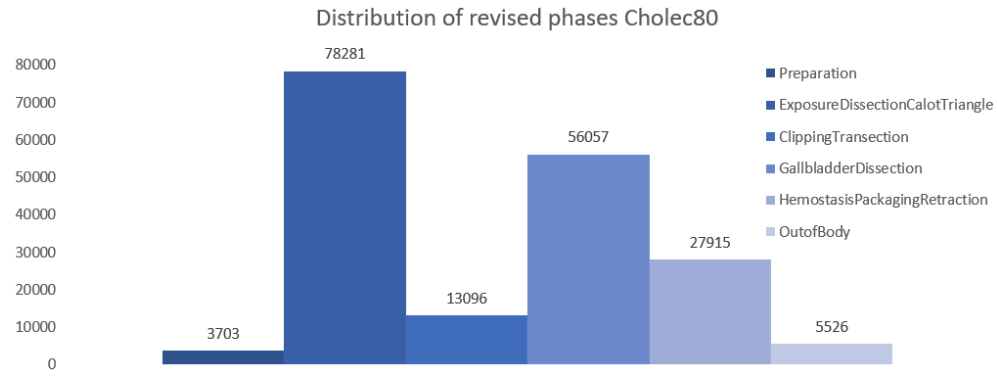

Figure 3.11: Distribution of the frames over the revised phases annotation of Cholec80 at one fps.

3.4.2 Evaluation of trained PD-network on original Cholec80 dataset

Colour-coded barplots

The trained network is applied to several videos of the test set for the conversion of the classified phases in colour-coded barplots. The results for video 72 - 75 of the test set are shown in figure 3.12. The top bar visualises the colour-coded classifications of the network for each frame, the middle bar shows the post processed result after filtering with a moving window and at the bottom the ground truth is shown with the frame numbers on the axis. The legenda shows the colour that corresponds to the phase number, as indicated in table 3.1.

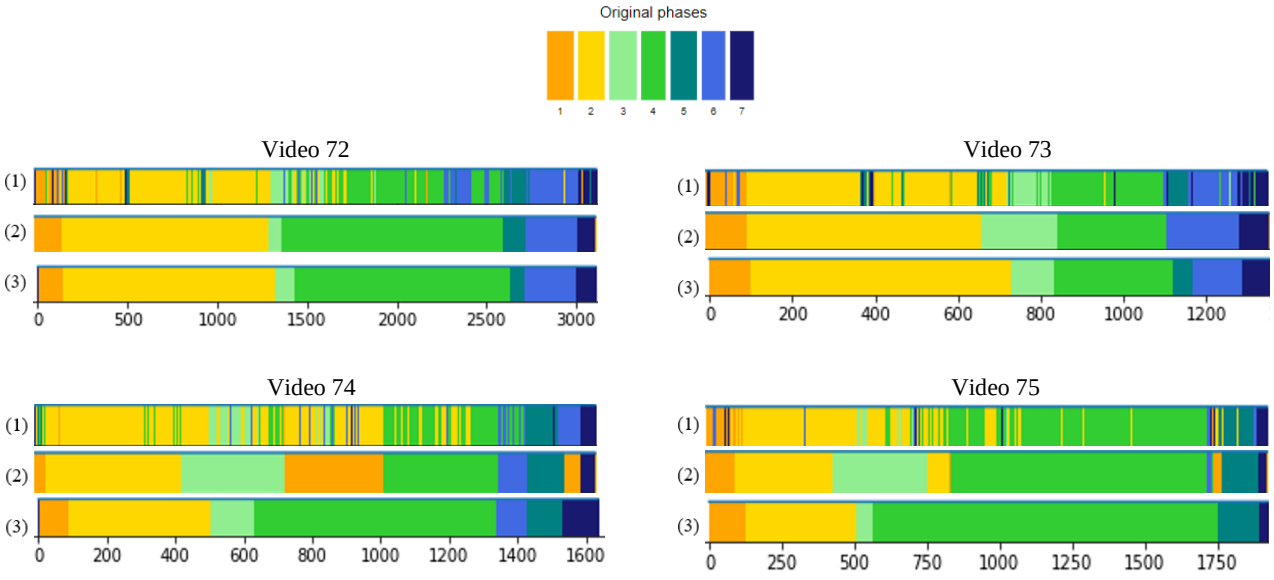

Figure 3.12: Colour barplots of the network classification (1), post processed (2) and ground truth (3) results for the original phase annotations on video 72, 73, 74 and 75 of the test set. The frame numbers of the video are plotted on the x-axis and the colour coded phase plotted on the y-axis.

Confusion matrices

The trained network is applied to the 32 videos of the test set for performance evaluation. The classification output for all frames of the test set is shown in the right CM and the normalised out in the left CM of figure 3.13. The normalisation is applied on the rows of the CM, the values of each row add up to 1.00. The colour-coded squares show a diagonal from top left corner to right bottom corner in the left CM and the right CM shows that the exact number of frames for each phase. The values on the diagonal of the left CM are the same as the recall values of the individual phases.

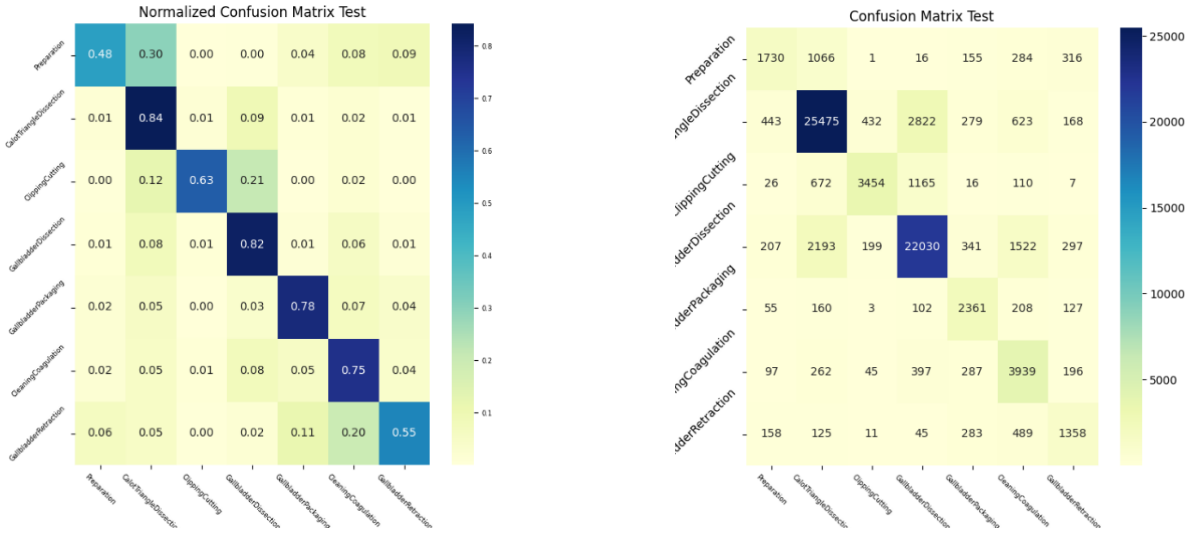

Figure 3.13: The normalized and absolute confusion matrices of the original phase annotations on the test set. The true class labels, surgical phases, are on the y-axis and estimated class labels, surgical phases, on the x-axis.

Performance metrics

The performance metrics used to evaluate the network in training and testing are the accuracy, precision, recall and F1-score. The validation performance of the network during training, after reaching peak performance and before converging, is presented in table 3.4. The test performance of the network during testing with the trained network is outlined in table 3.5. The segmental performance metrics for the individual phases of the test set is given for the precision, recall and F1-score, as the accuracy is an overall measure for the entire dataset. The network performs on the validation and test the highest in terms of precision from all performance metrics, 88.6% and 80.5% respectively. The difference in performance on the validation and test set is 2.9% in accuracy, 8.1% in precision, 8.0 % in recall and 8.0% in F1-score. The segmental metrics of test set show the individual performance. The majority phases, CalotTriangleDissection and GallbladderDissection, have the highest score in all segmental metrics and the minority phases substantially lower, with Preparation and GallbladderRetraction as lowest.

TABLE 3.4

VALIDATION PERFORMANCE METRICS ORIGINAL PHASE ANNOTATION

Phases	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Overall	***81.9 ± 0.7***	***88.6 ± 3.7***	***86.1 ± 5.4***	***87.3 ± 0.6***

Table 3.4: Performance metrics of the original phase annotation on the validation set with mean and std.

TABLE 3.5

TEST PERFORMANCE METRICS ORIGINAL PHASE ANNOTATION

Phases	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Overall	***79.0***	***80.5***	***78.1***	***79.3***
Preparation	-	66.4	48.3	55.9
CalotTriangleDissection	-	88.2	84.4	85.8
ClippingCutting	-	83.1	62.5	73.3
GallbladderDissection	-	83.4	81.5	82.7
GallbladderPackaging	-	62.9	77.5	71.2
CleaningCoagulation	-	55.4	74.6	65.1
GallbladderRetraction	-	55.3	55.2	55.3

Table 3.5: Performance metrics of original phase annotation on the test set.

3.4.3 Evaluation of trained PD-network on revised Cholec80 dataset

Colour-coded barplots

The trained network is applied to several videos of the test set, for the conversion of the classified phases in colour-coded barplots. The results for video 72 -75 of the test set are shown in figure 3.14. The top bar visualises the colour-coded classifications of the network for each frame, the middle bar shows the post processed result after filtering with a moving window and at the bottom the ground truth is indicated with the frame numbers on the axis. The legenda shows the colour that corresponds to the phase number as indicated in table 3.1.

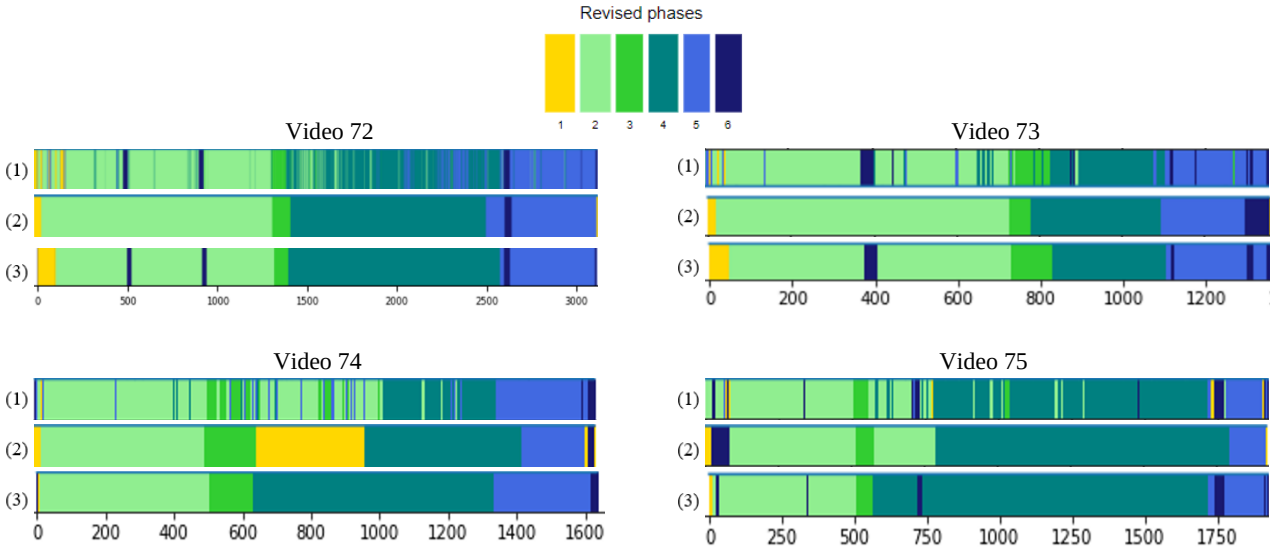

Figure 3.14: Colour barplots of the network network classification (1), post processed (2) and ground truth (3) results for the revised phase annotations on video 72, 73, 74 and 75 of the test set. The frame numbers of the video are plotted on the x-axis and the colour coded phase plotted on the y-axis.

Confusion matrices

The trained network is applied to the 32 videos of the test set for performance evaluation. The classification output for all frames of the test set is shown in the right CM and the normalised out in the left CM of figure 3.15. The normalisation is applied on the rows of the CM, the values of each row add up to 1.00. The colour-coded squares show a diagonal from top left corner to right bottom corner in the left CM and the right CM shows that the exact number of frames for each phase. The values on the diagonal of the left CM are the same as the recall values of the individual phases.

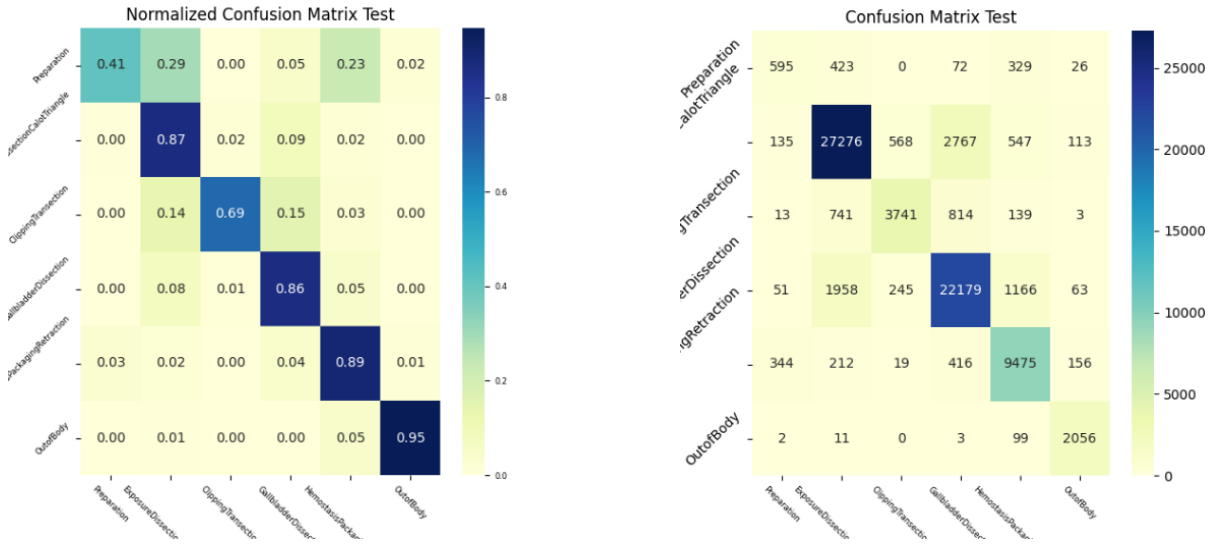

Figure 3.15: The normalized and absolute confusion matrices of the revised phase annotations on the test set. The true class labels, phases, are on the y-axis and estimated class labels, phases, on the x-axis.

Performance metrics

The performance metrics used to evaluate the network in training and testing are the accuracy, precision, recall and F1-score. The validation performance of the network during training, after reaching peak performance and before converging, is presented in table 3.6. The test performance of the network during testing with the trained network is outlined in table 3.7. The segmental performance metrics for the individual phases is given for all performance metrics except the accuracy. The difference in performance on the validation and test set is 3.2% in accuracy, 6.2% in precision, 5.1 % in recall and 5.2% in F1-score. The segmental metrics of test set show the individual performance. The ExposureDissectionCalotTriangle phase have the highest score on precision with 89.2% and OutofBody phase on recall with 95%. The Preparation phase scores substantially lower that the other phases in all segmental metrics.

TABLE 3.6

VALIDATION PERFORMANCE METRICS REVISED PHASE ANNOTATION

Phases	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Overall	***88.2 ± 0.8***	***92.5 ± 2.4***	***89.4 ± 2.1***	***90.5 ± 1.1***

Table 3.6: Performance metrics of revised phase annotation on validation set with mean and std.

TABLE 3.7

TEST PERFORMANCE METRICS REVISED PHASE ANNOTATION

Phases	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Overall	***85.0***	***86.3***	***84.3***	***85.3***
Preparation	-	52.2	40.8	46.0
ExposureDissectionCalotTriangle	-	89.2	87.3	88.3
ClippingTransection	-	82.3	69.4	75.4
GallbladderDissection	-	84.3	86.3	85.3
HemostasisPackagingRetraction	-	80.6	89.2	85.4
OutofBody	-	84.7	95.0	89.8

Table 3.7: Performance metrics of revised phase annotation on test set.

CAM images

The CAM overlay of the trained network on the input images, shows the discriminative regions of the image that are used to classify the (surgical) phase. The input image from the 72th video of the Cholec80 dataset and the resulting CAM images of the networks trained on the original and revised annotations are shown in figure 3.16, with the true and predicted phase by the networks. The CAM images of the first input image show that for the original annotations the bright white surface of the trocar is of interest and for the revised annotations more details of the trocar. The predicted GallbladderPackaging phase by the network for the original annotations is not the true phase. For the revised annotations, the predicted OutofBody phase is the same as the true phase of the input image. The CAM images of the second input image show that the networks trained on both annotations do not focus on the blue parts of the bipolar instrument. The CAM image for the original annotations has a large focus area at the top of the image, which includes the coagulated hepatic plate. The CAM image for the revised annotations has a clear focal point on the coagulated hepatic plate. The predicted CalotTriangleDissection phase by the network for the original annotations is not the true phase. For the revised annotations, the predicted HemostasisPackagingRetraction phase is the same as the true phase of the input image.

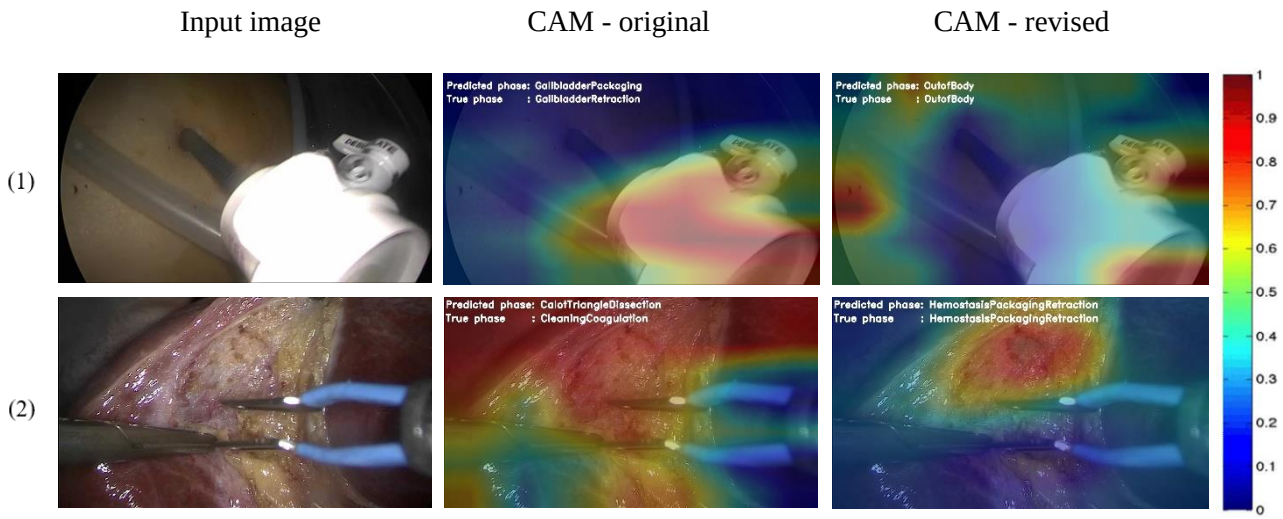

Figure 3.16: The input and CAM images from video 72 of the test set for the networks trained on original and revised annotations, with the predicted phase by the network and true phase of the input image.

3.5 Discussion

3.5.1 Research question and aim

The aim of this study was to develop a DL-network that can classify the surgical phases of the LC procedure on frames from intraoperative laparoscopic videos. Another aim of this study, was to investigate the importance of adequate labelling for detecting surgical phases of the LC procedure. The performance of a network is affected by both the network structure and the data. Most studies focus only on the development of their networks, rather than analysing their data. In this study, the Cholec80 data is analysed and reannotated. The effect on the network performance of the revised annotations is compared to the original annotations. The study results are discussed below.

3.5.2 Explanation of results

Data preparation

The distribution of the frames over the phases is imbalanced for the original and revised phase annotation datasets, as shown in figure 3.9 and 3.11. Both annotations have two clear majority and multiple minority phases. This not ideal for training purposes, as it introduces a bias towards the majority phases. However, this is a known aspect of clinical datasets and there are methods as class weighting or over-sampling that can be applied. The comparison of the revised annotations with the original shows the following aspects. The Preparation phase is reduced by 23%, as a result of the defined end point of the phase. The original phase had no clear description of the end point which resulted in fluctuation of the annotations over the videos in the dataset. The ExposeDissectionCalotTriangle is increased by 4%, ClippingTransection is increased by 2% and GallbladderDissection reduced by 5%. These minor changes could be the result of more strict annotation guide definitions. The HemostasisPackagingRetraction phase combines the last surgical phases, containing 13% less frames compared to the original phases. This can be the result of the action CleaningCoagulation, which was annotated in the original annotations as a phase. In the revised annotations, cleaning during phases is taken part of the surgical phases. The OutofBody phase is created from frames of all seven phases of the original annotation, containing 3% of the total number of frames.

PD-network on original Cholec80 dataset

Colour-coded barplots

The barplots of figure 3.12 show that the classifications of the network are noisy, meaning that individual frames are misclassified. This is a common feature of CNNs in classifying classes. Some frames of a phase are not discriminative for that phase, examples are idle time between phases or during the transition of surgical tools. The post processed result shows that the noise is removed and clear transitions of the phases can be distinguished. In video 72 and 73, post processing has resulted in clear phase blocks that are ordered in chronological order. These results resemble the ground truth in relatively high extent, with some minor deviations in the begin and end point of the phases. In video 74 and 75, it can be seen that the moving window has affected the length of phase blocks or even introduced new blocks. In video 74, after the block of the ClippingCutting phase a second Preparation block has been introduced. On the same location in video 75, a block of the CalotTriangleDissection has been introduced. This can be seen as an

artifact of post processing as each surgical phase occurs only once. In all four videos, the network misclassifies the phases of the frames that are annotated in the revised annotations as OutofBody. The network classifies a wide variety of phases for these frames, indicating that the network can not find discriminative features in these frames that correspond to the surgical phases. These frames introduce a standard error in the performance of the network.

Confusion matrices

The matrices of figure 3.13 show a diagonal from the left top to the right bottom, indicating that most of the frames are classified as the correct phase. The normalised values of the phases are 0.48 for Preparation, 0.84 for CalotTriangleDissection, 0.63 for ClippingCutting, 0.82 for GallbladderDissection, 0.78 for GallbladderPackaging, 0.75 for CleaningCoagulation and for GallbladderRetraction 0.55. The two majority phases, CalotTriangleDissection and GallbladderDissection, have the highest scores. That is to be expected, as there is more information to train on and there are more samples in the test data. Preparation, ClippingCutting and GallbladderRetraction are the three phase that have the lowest values. The frames of the Preparation phase are most misclassified as CalotTriangleDissection. As consecutive phases, the frames around the transition between the phases are most similar as the anatomy is still the same and the same tool, gasper, is often used. For ClippingCutting, most frames are misclassified as GallbladderDissection and CalotTriangleDissection. The clipping tool and scissors are the most discriminative features of this phase. When the tools are out of view, for instance to load new clips, the frames show great resemblance with these phases. As for GallbladderRetraction, most frames are misclassified as GallbladderPackaging and CleangingCoagulation. During all these phases the gallbladder retrieval bag has been introduced in the AC and might confuse the network.

Performance metrics

The results of this study are lower in terms of accuracy but for precision and recall compared to the results described by Czempiel et al. for the ResNet50 on the Cholec80.[71] Czempiel showed an accuracy of 82.2%, precision of 70.7% and recall of 75.9%. The network is trained until it converges, the optimal hyperparameter settings are used and the same data configuration is applied. The only explanation for the difference in performance is that in the study of Czempiel other videos of the Cholec80 were in the train, validation and test set, as this information was not published. However, the results of this study improved 4% in accuracy, 10% in precision and 12% in recall compared to the EndoNet described by Twinanda et al.[30] The test results of table 3.5 are reduced by 3%, 8.1%, 5.6% and 7.0%, for accuracy, precision, recall and F1-score respectively compared to validation. The difference in performance is to be expected as information of the validation set "leaks" into the network during training as the network weights are adjusted based on the validation performance. The network performs the highest on precision for the validation and test, with 88.6% and 80.5% respectively. This indicates that the network output has a high relevancy and low false positive rate of the classified phases. The segmental performance of the network on the test set is shown in table 3.5. All phases have precision and recall scores that are within a close range of each other, except for the Preparation phase. The precision of this phase is 66.4% and recall 48.3%. The latter, indicates that less of the frames are classified as Preparation, but of the frames that are classified as Preparation, a higher amount is correct. However, the GallbladderRetraction phase is the worst performing phase with a F1-score of 55.3%. The CalotTriangleDissection scores the highest in all segmental metrics with precision of 88.2% and recall 84.4%. That indicates that most of the frames classified as CalotTriangleDissection and most of the frames that are classified as CalotTriangleDissection are correct.

PD-network on revised Cholec80 dataset

Colour-coded barplots

The barplots of figure 3.14 show that the classifications of the network are also noisy and individual frames are misclassified. Using the same network, this common feature of CNNs in classifying classes was also expected. The noise is introduced by frames of a phase are not discriminative for that phase. These frames are caused by idle time between phases or during the transition of surgical tools for example. The post processed result shows that the noise is removed and clear transitions of the phases can be distinguished. In video 72 and 73, post processing has resulted in clear phase blocks that are ordered in chronological order and resemble the ground truth to a high extent. However, some of the OutofBody phase blocks are filtered out. This artifact has no effect on detection of the phase transitions of the surgical phases. The surgical phase blocks show some minor deviations in the begin and end point of the phases, in respect to the ground truth. In video 74 and 75, it can be seen that the moving window has affected the length of phase blocks or even introduced new blocks. In video 74, after the block of the ClippingTransection phase a second Preparation block has been introduced. On the same location in video 75, a block of the ExposeDissectionCalotTriangle has been introduced. This is a post processing artifact, as each surgical phase occurs only once. In all four videos, the network shows a high capability to classify the frames of the OutofBody phase. This indicates that these frames should not be annotated as surgical phases. These results prove that the annotation of the frames outside the abdominal cavity as a separate class removes the standard error of the network's classifications and thereby improves the performance.

Confusion matrices

The matrices of figure 3.15 show a diagonal from the left top to the right bottom, indicating that most of the frames are classified as the correct phase. The normalised values of the phases are 0.41 for Preparation, 0.87 for ExposueDissectionCalotTriangle, 0.69 for ClippingTransection, 0.86 for GallbladderDissection, 0.89 for HemostasisPackagingRetraction and for OutofBody 0.95. All surgical phases, show an improvement compared to the performance with the original annotations except for the preparation phase. The reduce in performance can be explained by the 23% reduction in frames. In contradiction to the original annotations, the two majority phases do not have the highest scores in the revised annotations. The phases with the highest scores are the newly defined phases HemostasisPackagingRetraction and OutofBody with 0.89 and 0.95 respectively. This indicates that the revised phase definitions improve trainability and performance. The majority phases do have high scores, indicating that their performance has not be negatively affected by class weighting. Preparation and ClippingTransection phase have the lowest values. The frames of the Preparation phase are most misclassified as ExposureDissectionCalotTriangle and HemostasisPackagingRetraction. As consecutive and majority phase, it is to be expected that most frames are misclassified as ExposureDissectionCalotTriangle. The frames around the transition between the phases are most similar as the anatomy is still the same and the same tool, gasper, is often used. The high number of misclassifications for HemostasisPackagingRetraction, 0.23, are the comparable with the misclassifications of the separate phases of the original annotation, being 0.04, 0.08 and 0.09. For ClippingTransection, most frames are misclassified as GallbladderDissection and ExposureDissectionCalotTriangle. The clipping tool and scissors are the most discriminative features of this phase. When the tools are out of view, for instance to load new clips, the frames show great resemblance with these phases.

Performance metrics

The network shows improved performance on the revised annotations compared to the original in table 3.6 and 3.7. The network performance of the revised annotations on the test set is improved by 6.0%, 5.8%, 6.2% and 6.0%, for accuracy, precision, recall and F1-score, respectively, compared to the original annotations. The original and revised annotations have seven and six phases respectively. In order to compare the overall performance metrics, a simple correction for the difference in guess chance could give an indication or more advanced statistical analysis has be conducted by a Monte-Carlo simulation. In this study the guess chance correction is used to give an indication of the corrected performance difference and limit the computational burden. The guess chance with seven classes is (1 / 7) * 100% = 14.3% and for six (1/ 6) * 100% = 16.7%. The difference in guess chance is 2.4%, which is the correction factor for the performance metrics. After correction, the improvement of the revised annotations over the original are 3.6%, 3.4%, 3.8% and 3.6%, for accuracy, precision, recall and F1-score, respectively. The revised test results of table 3.7 are reduced by 3.2%, 6.2%, 5.1 % and 5.2%, for accuracy, precision, recall and F1-score, respectively, compared to validation. The difference in performance is also a result of information of the validation set "leaking" into the network and is comparable with the results for the original annotations. The network performs the highest on precision for the validation and test, with 92.5% and 86.3% respectively. That also indicates that the network output has a high relevancy and low false positive rate of the classified phases. The segmental performance of the network on the test set is shown in table 3.7. The Preparation and ClippingTransaction phase have both a substantial difference between the precision and recall score. The precision for both phases is higher than the recall, which indicates that less of the frames are classified as the phase but of the frames that are classified, a higher amount is correct. However, also for the revised annotation the Preparation phase is the worst performing phase with a F1-score of 46.0%. The performance of the Preparation phase is declined for the revised annotations compared to the original, which is probably the result of the 23% reduction in frames. The OutofBody scores the highest on recall with 95.0%, almost all frames are returned are truly relevant. That indicates that the visual features of these frames are distinctive from the other phases. ExposureDisscetionCalotTriangle scores best over all segmental metrics with 89.2% precision, 87.3% recall and a F1-score of 88.3%. This indicates that most of the frames are classified as ExposureDisscetionCalotTriangle and most are correct. The performance of the HemostasisPackagingRetraction phase has drastically improved compared to the individual CleaningCoagualtion, GallbladderPackaging and GallbladderRetraction phase of the original annotations. The results for HemostasisPackagingRetraction are 80.6% precision, 89.2% recall and a F1-score of 85.4%. For the phases of the original annotation, the precision ranges from 55.3% - 62.9%, the recall from 55.2% - 77.5% and for the F1-score from 55.3% - 71.2%.

CAM images

The CAM images in figure 3.16 show the difference in focus regions of the networks trained on the original and revised annotations. For the first input image of figure 3.16, the CAM of the original annotations shows that the network relates the bright white colour of the trocar to the GallbladderPackaging phase. In this phase, the gallbladder retrieval bag is introduces in the AC which also has a white colour. The network mistakes the trocar for the retrieval bag and misclassifies the frame as GallbladderPackaging, instead of GallbladderRetraction. The network struggles to classify frames outside the AC with the original annotations as these do not contain information related to the surgical phases. The CAM of the revised annotations shows that the network focusses on the details of the trocar, as the insufflation valve and opening of the trocar. The network relates these details to the OutofBody phase, which is the correct phase.

The introduction of this phase provides the network the opportunity to define features that are specific for frames outside the AC. For the second input image of figure 3.16, the CAM of the original annotations shows that the network does not focus on the surgical tools, grasper and bipolar. The grasper is a tool that is present in many phases but the bipolar only in the CleaningCoagulation phase. This might indicate that this phase is to diverse in order to define the correct features on. The network classifies this frame as CalotTriangleDissection and the CAM shows that the network focuses on the complete coagulated hepatic plate. The shape and colour of the plate resembles the gallbladder with cystic pedicel, hence the misclassification. The CAM of the revised annotations shows that the network focusses specifically on the coagulated hepatic plate with a more focussed region than the original annotations. The surgical tools are also for this network not discriminative, as the bright blue colour of the bipolar is quite distinctive. The bipolar is only use in some videos, so might therefore be not generalisable enough. The network correlates the coagulated plate with the HemostasisPackagingRetraction phase, also being the true phase. The CAM images of both networks did not show any biases in the visual data of the Cholec80 dataset.

3.6 Conclusion

This chapter described the first part of this study, which aimed to development a DL network that can accurately and objectively classify the surgical phases of intraoperative LC videos. It can be concluded that it is possible to objectively classify surgical phases with a base-line CNN and reach comparable performance, as stated in other research. The evaluation of the revised annotations with the original annotations of the Cholec80 LC dataset, showed that the network performance improved by removing the standard error in the data. The performance metrics indicated that the revised annotations improved 6.0%, 5.8%, 6.2% and 6.0%, for accuracy, precision, recall and F1-score respectively. The HemostasisPackagingRetraction phase showed an improvement between 14.2% and 30.1% on the F1-score compared to the last three phases of the original annotation. The OutofBody phase scored outstanding with 84.7% precision, 95.0% recall and a F1-score of 89.8%, especially as it only contains 3% of the frames in the dataset. The CAM images provided insight into the network's regions of interest. For the revised annotations, the focus was more centred in the view of the laparoscope and located around key structures compared to the original annotations. These results give an indication about the clinical importance of adequate labelling, in surgical phase classification of LC video data. The noisy character of the CNN classification results could be reduced by post processing with a moving window filter. Clear phase transitions were distinguishable in the post processed phase output. However, fixed filter settings resulted in inconsistent processing results and introduction of artifacts. The proposed solution for further research is the use of a TCN for the classification of the phases. A TCN has no noisy classification character and therefore does not need post processing before the detection of phase transitions.

CHAPTER 4

4. Predict remaining laparoscopic cholecystectomy procedure duration

This chapter describes the second part of this study, which is the development a ML network that can accurately and objectively predict the remaining LC procedure duration and update the prediction after each phase. The duration of the individual surgical phases of intraoperative laparoscopic videos provided in the first part of this study (see Chapter 3) is used as input for the network to predict and update the remaining procedure duration.

4.1 Introduction

4.1.1 Prediction of remaining surgical procedure duration

As the department of surgery is one of the busiest hospital units, optimal scheduling of procedures is essential to maximize the utility of the surgical facility resources. This creates the need for accurate predictions of total and remaining surgery duration. In current clinical practice, the preoperative predicted surgery duration is based on average durations and rough estimations. During the day, schedulers try to dynamically adapt the OR schedule based on the progress of the individual operations. Therefore, typically verbal communication with the OR staff is used to obtain estimates of the remaining procedure time (RPT). The first disadvantage of the current method is the disruption of the workflow on the OR, which might even compromise the safety of the patient and personnel. The second disadvantage is unforeseen prolonged operating time as a result of duration underestimation. This is the main reason for surgery cancellations due to a lack of OR availability. Surgery delay or cancellation increase the preoperative waiting time for patients and the overtime for OR personnel. The third disadvantage is the higher expenditure of the OR due to underutilization of the resources in terms of increased idle time, overtime and rescheduling as a result of over- or underestimation of surgical procedure duration. In addition, interactive timetables that use all the available information could also improve patient safety in terms of reduced duration of anaesthesia, ventilation, and intensive care. However, the incorporation of all the available information is difficult for OR schedulers due the variability of the procedure duration caused by a high diversity of patients, surgeons, and intraoperative situations.[75] The development of automated scheduling tools provide the possibility to incorporate all the available information for the scheduling process, without disturbing the OR personnel, and make accurate predictions. Improvement on the accuracy of procedure duration predictions would result in better arrangement of surgical procedures throughout the ORs. This results in more efficient use of the resources, which reduces the costs and increase the revenue by allowing more surgeries to be performed.

4.2 Technical background

4.2.1 Regression models

The models used to estimate the time need to apply regression techniques because the procedure duration is a continuous variable, i.e. time. The most common regression techniques are SLR and MLR. SLR and MLR are fitting a linear model with one or multiple coefficients between one or more input vectors, phase durations, and a dependent output variable, RPT. LRs try to minimize the residual sum of squares by linear approximation.[76-78] . A RF regression applies ensemble learning by combining classifying DTs in a random process on multiple sub-sample sets of the complete dataset. RF takes the advantage of the predictive power from each DT. The trees can use MSE, MAE or the Poisson as an optimization criterion. Each tree is grown on a bootstrap sample, random sampling with replacement, of the training cases and the tree node splits based on a random subset of the input variables. The RF regression calculates an unweighted average over all trees for the prediction, improving the predictive accuracy and control on over-fitting.[77] [78] SVR can apply either a linear, polynomial, radial basis or sigmoid function to solve a regression problem. SVR aims to minimize the sum of squared errors and therefore uses an additional penalty parameter. The error term is handled by the penalty parameter outside the specified error margin. The regression model updates an initial value for the average procedure duration after each phase. The most value for the OR schedulers is in the updated predictions after the first three or four phases because the remaining procedure time is long enough to make alternations in the OR schedule.

4.2.2 K-fold cross-validation

The data used for ML are partitioned into a train, validation and test set for training, hyperparameter tuning and performance evaluation purposes. The model is trained on the train set, tuned on the validation set and the performance is evaluated on the test set. Tuning the hyperparameters on the validation set prevents the risk of overfitting on the test set. The estimator could else be tweaked until optimal performance, in that case knowledge about the test set “leaks” into the model. The performance metrics then no longer resemble the generalised performance. The variance is the dataset creates an uncertainty in the performance metric score. K-fold cross-validation is a partition technique that is applied for accurate evaluation of the model. It divides the data sample used for training and validation into k subsamples of equal size. One subsample is retained for validation of the model and the remaining subsamples are used for training the model. This cross-validation process is repeated k times and the evaluation results of all folds are averaged for a mean performance estimation. The test set is held for the final model performance evaluation. The K-fold cross-validation approach is more computationally extensive but limits the size of the validation set. This is a major advantage when the dataset is relatively small.[79] [80] Figure 4.1 shows a visual representation of K-fold cross-validation on the training data with five folds. The test data is held for final performance evaluation.

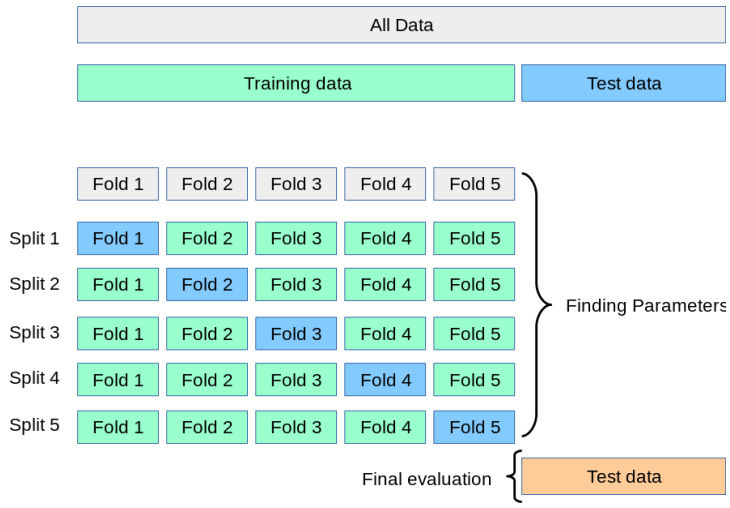

Figure 4.1: K-fold cross-validation on training data, K is five. Performance evaluation of model on test data.[79]

4.2.3 Model performance

The performance evaluation of the regression model for the prediction of the procedure duration, will be assessed by comparing the prediction with the actual duration. The prediction could also be compared to the estimation made by the OR schedulers of the MMC when available in the data acquisition of the new data. A regularly used performance evaluation metrics for regression models is the Root Mean Square Error (RMSE).[77] [78] [81] [82] The advantage of RMSE is that it personalizes variance by applying more weight to the errors with larger absolute values. Hence the RMSE tends to become increasingly larger for increasing variation in the distribution of the error magnitudes.[80] The formula to calculate the RMSE is given by equation 4.1 wherein n is the number of samples, *i* is the sample number starting at one, *f(x)* is the predicted value and *y* is the actual value of the estimated variable for the given sample number. The subtraction of *f(x)* from *y* gives the error between the predicted and actual value. The errors of each sample in the sample size are squared. The squared errors are summed and divided by the sample size to calculate the mean squared error (MSE). Finally, the root is taken of the MSE for the RMSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2}$$

Equation 4.1: The function for the RMSE.[77] [81] [82]

In addition to the RMSE, the coefficient of determination (R^2) is used as a performance score for regression models. The article of Chicco et al. from 2021 in the PeerJ Computer Science journal stated that the R^2 should be a standard metric for the evaluation of regression analysis. The R^2 does not have the interpretability limitations of the MAPE, MSE, and RMSE.[83] The R^2 expresses the proportion of the variation in the predicted variable(s) based on the independent variable(s). It provides information about the quality of the fit of a model to the data. The R^2 normally ranges from zero to one but can also become negative. The best model performance is achieved by $R^2 = 1$ and baseline model performance is given by $R^2 = 0$. Negative R^2s resemble worse predictions than the baseline. In this case, the mean of the data forms a better fit than the predicted values. Indicating that the model does not fit to the data. Only R^2 values

between zero and one can be evaluated for the model performance. The R^2 is calculated over the n values of the dataset ($y_1 ... y_n$) and the associated predicted values of the model ($f_1 ... f_n$). The variability of the dataset can be measured with the residual sum of squares (SS_{res}) and total sum of squares (SS_{tot}), given by function 1 and 2 of equation 4.2. The R^2 is calculated with the SS_{res} and SS_{tot} as shown in function 3 of equation 4.2.[84]

$$SS_{res} = \sum_i (y_i - f_i)^2 \quad (1)$$

$$SS_{tot} = \sum_i (y_i - \bar{y}_i)^2 \quad (2)$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (3)$$

Equation 4.2: The functions for SS_{res}, SS_{tot} and the R^2.[84]

4.2.4 Previous research

Several studies proposed approaches to address the current problem of OR scheduling. These approaches focussed on predicting the surgery duration preoperatively. For instance, predictions of the surgery duration based on the historical data about the procedure and the surgeon.[85] Other data, such as the age of the patient, the OR, the OR team, the day of the week, the month, and the year have also been investigated.[86] Such preoperative approaches still have a difficulty dealing with the unpredictability and uniqueness of each surgical procedure. In some studies, semi-automatic methods are proposed that require input of anaesthesiologists during the procedure.[87] Similar workflows are used in most hospital in current clinical practice due to a lack of reliable automated systems. These semi-automatic approaches are not desirable as they disrupt the processes on the OR. Guédon et al. used the activation signal of electrosurgical devices as input signal to determine when the next patient should be ordered. The limitations of the proposed pipeline are that the detection signal started 15 min in the procedure and was based on the assumption that preparation of the next patient should be started 25 min before the end of the procedure.[88] These limitations indicate the method can not be applied on a wider variety of surgical procedures. Multiple studies investigated the possibility to give updates about the progress of procedures and the capability of making reliable predictions of the procedure duration. These studies used predictive modelling approaches as SLR and RF.[76] [77] ShahabiKargar et al. compared the performance of the SLR and RF with the hospital estimation of multiple procedures. The SLR showed a 0.9% overall shortcoming compared to the hospital estimate and the RF showed a 28% overall improvement on the hospital estimate.[77] In a later study, ShahabiKargar et al. showed that after filtering the unreliable data and applying new ensemble approaches, the RF had an improvement of 44% on the hospital estimate.[78] Twinanda et al. was the first study to solely use visual data of 120 LC procedures as an input for their models to predict the RPT. They applied regression on the output of a CNN-LSTM by the LSTM, showing a Mean Absolute Error (MAE) of 15.6 min.[75] Later, Bodenstedt et al. investigated the use of visual data from 80 variating laparoscopic procedures. They also used a CNN-LSTM for real-time prediction of the RPT, showing MAE of 36.7 min.[89] Both studies use CNN-LSTMs to predict the RPT based on the spatial and temporal information of the LC video data. There is no study that investigated the use of regression models to predict the RPT based on solely temporal information, in terms of the surgical phase durations of the LC procedure.

4.3 Materials and Methods

4.3.1 Intraoperative phase duration dataset

The source data consist of the publicly available Cholec80 dataset with intraoperative LC videos used to train the previously described DL network in the prediction of the phases. The revised phase definition is used to create the phase duration dataset, for their more suited clinical and technical relevance. The five surgical phases are Preparation, Exposure and Dissection of Calot's Triangle, Clipping and Transection of cystic duct and artery, Gallbladder Dissection from fossa/hepatic plate, and Hemostasis, Packaging and Retraction of gallbladder. The last revised surgical phase, HemostasisPackagingRetraction, combines the short individual phases; GallbladderPackaging, CleaningCoagulation and GallbladderRetraction at the end of the procedure in the original annotation of the Cholec80. These individual phases impose technical difficulty and limited value in the prediction of the RPT, as they are located at the end of the procedure. CleaningCoagulation is technically an action and not a phase. It is therefore not present in every video, which causes inconsistency that might confuse the prediction model. This action is incorporated in the surgical phases of the revised annotations. The Preparation phase is also a phase that in inconsistent in the Cholec80 dataset, due to delayed recording of the intraoperative video data by the OR personnel. The revised annotation guide is made based on the clinical and technical relevance of the phases in the LC procedure, with the intention to be used for new inclusions from the MMC in the future. Hence, the Preparation phase is included in the revised annotation guide despite being inconsistent in the Cholec80. As Hong et al. showed, more generalised annotation processes are preferred than specifically tailored definitions for each individual dataset.[35] The high variation in length and presence might affect the predictive value of this phase for the RPT, however. The phase duration dataset for training the predictive models, is derived manually from the revised phase annotations at one fps. The phase transitions of the five surgical phases are used to determine the RPT after the end of each phase. The number of frames can directly be related to the duration since the recording started. The duration of the first phase is subtracted from the total procedure time to derive the RPT. The duration of subsequent phases is subtracted from the RPT of the previous phase, updating the RPT after each phase. In videos with absence of Preparation due to delayed recording, a phase duration of zero seconds is used.

4.3.2 K-fold cross-validation

The phase duration dataset was split in a train, validation and test set with a ratio of 0.88 : 0.06 : 0.06 respectively. This resulted in 70 train, five validation and five test videos. The ratio was chosen to maximize the size and information of the train set for training purposes. The split was chosen so that the train, validation and test data had a comparable distribution over the phases. The phase durations were split per video, meaning that all phases of the same procedure were either in the train, validation or test set. K-fold cross-validation was used for hyperparameter optimisation. The validation dataset consisted of five videos and was variated five folds over the combined 75 videos of the train and validation dataset. The model weights of previous trainings were not transferred to consecutive training in the K-fold, in order to maintain that all models started from the same point. This ensured that the measured performance was a result of the obtained information from that specific data configuration and not from previous configurations.

4.3.3 Regression models implementation and training

The prediction of the RPT has been performed with three regression models: LR, RF and SVR for evaluation of best performance. All regression models are retrieved from the Scikit-learn library version 1.0.1, a free software library for ML in Python. The hyperparameters of the models were selected based on the features of the dataset and for some hyperparameter optimisation was performed. The LR model was trained with the following hyperparameter settings. Fit_intercept is True, the data is not expected to be centred. Normalize is False. N_jobs is None, can be set to set number for speeding up the computational time. Positive is False, the coefficients are not forced to be positive. Coef is n_features as the input is one-dimentional.[90]

The RF model was trained with the following hyperparameter settings. N_estimators was set to 100, the number of DT in the forest. Criterion is squared error, the MSE is used as optimisation criterion. Max_depth is None, the nodes in the tree will expand until all leaves are pure or the samples are smaller than min_samples_split. Min_sampels_split is two and min_samples_leaf is one, which is the minimal value for integers. Min_weight_fraction_leaf is 0.0, giving equal weight to all leaves. Max_features is auto, that set it equal to the n_features. Bootstrap is True. N_jobs is None, the trees are run one by one. Max_samples is None, one sample is used to train the base estimator.[91]

The SVR model was trained with the following hyperparameter settings. Kernel is poly, using the polynomial function. Degree is three, a third order polynomial. Gamma is scale, which used 1 / (n_features * variance) coef0 is 0.0, start value of the coefficients. Tol is $1 * e^{-3}$, tolerance value of the stopping criterion. C is 1000.0, the value of the L2 penalty. Epsilon is 0.1, the width of the tube in which no penalty is associated to the training loss for points that are off the actual value. Max_iter is infinite, the no limitation of iterations of the solver.[92]

4.3.4 Model performance evaluation

The model performance was evaluated five times with variation distribution of the Cholec80 videos over the split for cross-validation because of the small size of the test set. A small test set can introduce a bias in the performance of the model. The video data of that small sample might resemble the average of the total dataset or lay fare apart, effecting the parameter values. The performance was evaluated based on the RMSE and R^2 of the predicted and true RPT for all the surgical phases. The RMSE indicates the error between the predicted and actual value by subtraction. The root is taken of the errors, then they are averaged and squared, resulting in the absolute average error. The RMSE is presented both seconds and minutes.

The R^2 presents the proportion of the variation in the predicted RPT based on the variance in the true RPT. The R^2 shows the quality of the fit made by the model on the data, ranging from zero to one but can also become negative. The results of the model predictions are visualised in a graph for each video of the test set. The RPT in seconds is plotted against the surgical phases for the true RPT with an acceptance range of five min, the predicted RPT and the 45 min standardized preoperative estimate used in clinical practise at the MMC. The 45 min estimate is the baseline to evaluate the model performance to the clinical practice.

4.3.5 Statistical analysis

The statistical analysis to compare the fit of the regression models to the data is measured with the log likelihood function. The function expresses the estimation performance for a free variable parameter (θ) based on the observations. The log likelihood is preferred over the maximum likelihood, as it is simpler to compute and often easier to optimise.[93] For x samples of independent and identical distributed observations, their joint probability density, likelihood, function is presented by equation 4.3.

$$f(x_1, x_2, \dots, x_n|\theta) = f(x_1|\theta) * f(x_2|\theta) * \dots * f(x_n|\theta) = \prod_{i=1}^{n} f_X(x_i|\theta)$$

Equation 4.3: The joint probability density, likelihood, function for x observations and the parameter (θ).[93]

The log likelihood of the x samples and θ is given by equation 4.4, as the log of a product is represented by the summation of the logs of the individual product terms.

$$\ln f\,(x_1, x_2, \dots, x_n|\theta) = \sum_{i=1}^{n} \ln f(x_i|\,\theta)$$

Equation 4.4: The log likelihood function for x observations and the parameter (θ).[93]

The statistical significance of the differences between the predicted RPTs of the models is assessed by a log likelihood function with a confidence interval (CI) of 95%. The results are considered statistically significant with a P-value < 0.05. The statistical analysis was performed with the statistical software SPSS (IBM Corp. Released 2020. IBM SPSS Statistics for Windows, Version 27.0. Armonk, NY: IBM Corp.)

4.3.6 Remaining procedure time prediction pipeline

A pipeline has been developed for this study, shown in figure 4.3, to process the detected phase durations from the video data of the Cholec80 by the CNN, ResNet50, with the revised phase annotations. The phase duration dataset consists of the RPT in seconds, after the five surgical phases of the videos from the train and validation set. The detected phase durations by the CNN of the test set, are passed through the model after training for performance evaluation and the updated RPT after each phase. The predicted RPT is evaluated based on the true RPT, with a five min acceptance range, and the standardized initial estimate of 45 min used in the MMC. These parameters are plotted in a graph with the surgical phase on the x-axis and RPT on the y-axis. The true RPT is not a straight declining line due to the difference in phase duration of the surgical phases and the fact that the phases have the same length on the x-axis. Performance metrics are calculated based on the difference between the predicted and true RPT.

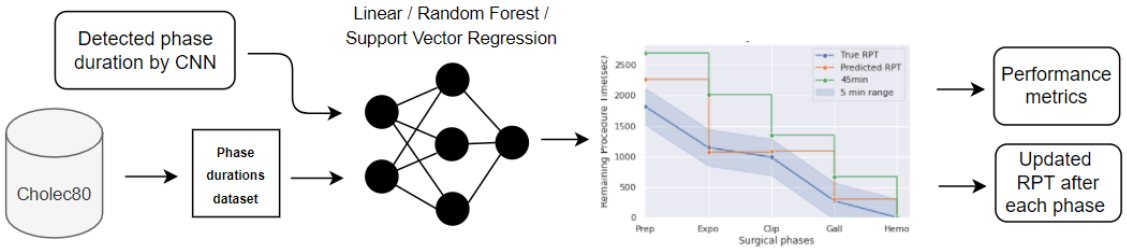

Figure 4.3: RPT prediction pipeline with the phase duration input, updated RPT after each phase and performance metrics output.

4.4 Results

4.4.1 Intraoperative phase duration dataset

The intraoperative phase duration dataset of the Cholec80 with revised annotations of the Cholec80, was sampled at one fps for this study. The phase durations of the entire dataset are shown on the top left of figure 4.4. The mean phase duration of each surgical phase in seconds with std is: Preparation 63 ± 206, ExposureDissectionCalotTriangle 998 ± 734, ClippingTransection 189 ± 167, GallbladderDissection 728 ± 645 and HemostasisPackagingRetraction 413 ± 217. The mean and std of the total procedure time is 2357 ± 976. The training dataset consists of 70 videos and has almost the exact same distribution of the phase durations over the surgical phases as the total dataset, shown in the on the top right of figure 4.4. The validation set consist out of five videos and has a distribution that comes close to the training dataset, shown in the on the bottom left of figure 4.4. The Preparation is 48 ± 110, ExposureDissectionCalotTriangle 814 ± 349, ClippingTransection 157 ± 66, GallbladderDissection 823 ± 335 and HemostasisPackagingRetraction 502 ± 279. The mean and std of the total procedure time is 2448 ± 793. The test set also consists of five videos and has a distribution that comes close to the validation dataset, shown in the on the bottom right of figure 4.4. The Preparation is 117 ± 121, ExposureDissectionCalotTriangle 819 ± 408, ClippingTransection 252 ± 249, GallbladderDissection 718 ± 224 and HemostasisPackagingRetraction 527 ± 236. The mean and std of the total procedure time is 2486 ± 769. K-fold cross-validation is applied on the train and validation set with five folds.

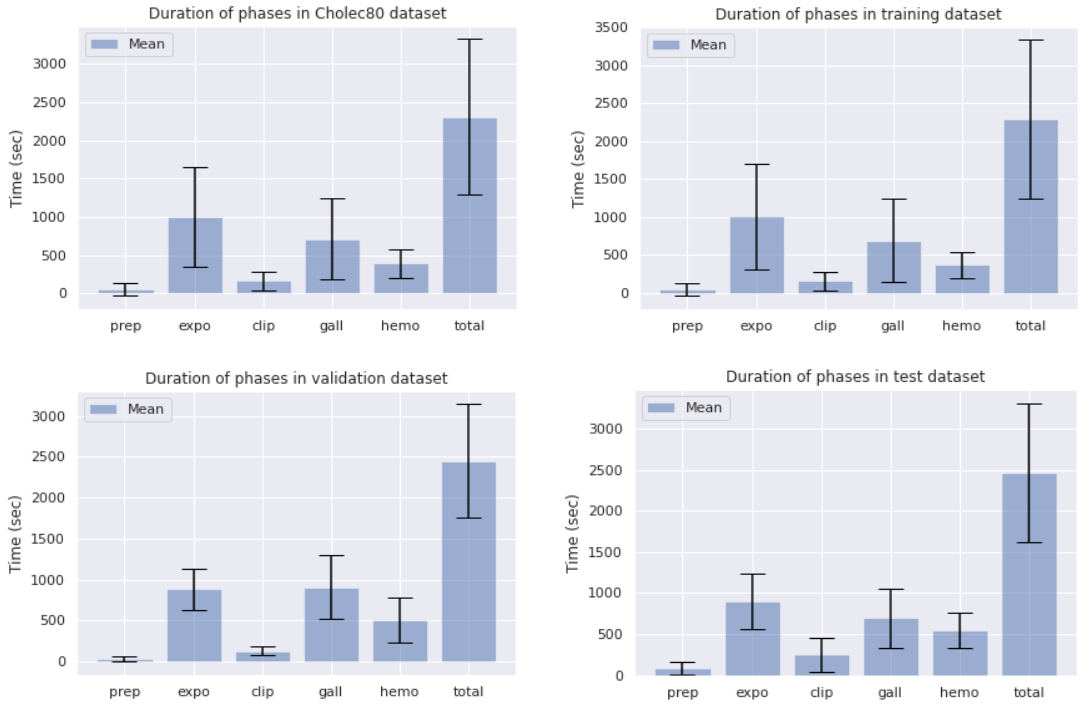

Figure 4.4: Deviation of the phase durations over all datasets for the five surgical phases and the total procedure time.

Linear Regression

The performance metrics used to evaluate the LR model in training and testing are the RMSE and R^2. The validation performance of the model during training is presented on top table 4.1 and for the test set on the bottom. The validation and test results are the averaged results over the five folds of training with cross-validation, so in total of 25 videos. The RMSE shows the overall absolute error of the RPT prediction for each surgical phase of that video. The RMSE is given in seconds and minutes. The R^2 expresses the proportion of the variation in the RPT predictions based on the variation in the phase durations dataset. The averaged validation performance for the fist video is the best with a RMSE of 411 ± 80 sec or 6.8 ± 1.3 min and R^2 of 0.6 ± 0.4. The overall score of all five videos is a RMSE of 558 ± 207 sec or 9.3 ± 3.5 min and R^2 of 0.4 ± 0.7. For the test set, the performance over the five folds is from the five same videos. The averaged test performance for the fourth video is the best with a RMSE of 305 ± 37 sec or 5.1 ± 0.6 min and R^2 of 0.9 ± 0.0. The overall score of all five videos is a RMSE of 605 ± 37 sec or 10.1 ± 0.6 min and R^2 of 0.3 ± 0.0. The third and fifth video of the test set show a negative R^2.

TABLE 4.1

PERFORMANCE METRICS LINEAR REGRESSION MODEL

Videos val set	**RMSE (s)**	**RMSE (min)**	**R^2**
1	411 ± 80	6.8 ± 1.3	0.6 ± 0.4
2	782 ± 64	13.0 ± 3.6	0.2 ± 0.5
3	533 ± 247	8.9 ± 4.1	0.2 ± 1.1
4	649 ± 367	10.8 ± 6.1	0.3 ± 0.8
5	415 ± 131	6.9 ± 2.2	0.5 ± 0.5
overall	558 ± 207	9.3 ± 3.5	0.4 ± 0.7

Videos test set	**RMSE (s)**	**RMSE (min)**	**R^2**
1	404 ± 19	6.7 ± 0.3	0.8 ± 0.0
2	697 ± 20	11.6 ± 0.3	0.4 ± 0.1
3	873 ± 16	14.6 ± 0.3	0.2 ± 0.0
4	305 ± 37	5.1 ± 0.6	0.9 ± 0.0
5	747 ± 94	12.5 ± 1.6	-0.7 ± 0.0
overall	605 ± 37	10.1 ± 0.6	0.3 ± 0.0

Table 4.1: Performance metrics of the five-fold cross-validated LR model with mean and std.

The results of the LR model are visualised in figure 4.5. The first three videos of the test set show, the predicted RPT after each surgical phase in relation to the true RPT with a five min acceptance range and the preoperative estimate of 45 min that is used as a standard in the MMC. The y-axis shows the RPT in sec and the x-axis the five surgical phases. The prediction is made after the surgical phase has ended. The first video shows a prediction within the five min range for all but the Preparation phase, the second and third only for the last two phases. In the first and third video, the predicted RPT is closer to the true RPT than the 45 min estimate. The high RMSE of video one, shown in table 4.1, results from the estimate for the Preparation phase. The R^2 is still high as the predictions for all other phases are close to the truth. Also for the second and third video, the RMSE results from the RPT prediction after the Preparation phase. The R^2 for the second video is 0.4 as the other predictions are in the range of the true values of the RPT and for the third 0.2 as they are not.

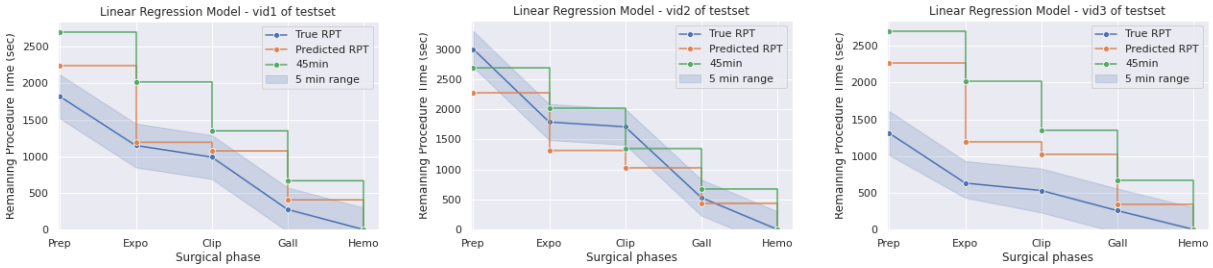

Figure 4.5: Graphical plots for LR model of the true, predicted RPT, five min range and 45 min estimate for the surgical phases.

Random-Forest regression

The performance metrics used to evaluate the RF model in training and testing are the RMSE and R^2. The validation performance of the model during training is presented on top table 4.2 and for the test set on the bottom. The validation and test results are the averaged results over the five folds of training with cross-validation, so in total of 25 videos. The averaged validation performance for the fifth video is the best with a RMSE of 376 ± 156 sec or 6.3 ± 2.6 min and R^2 of 0.6 ± 0.4. The overall score of all five videos is a RMSE of 454 ± 200 sec or 7.6 ± 3.3 min and R^2 of 0.5 ± 0.5. For the test set, the performance over the five folds is from the five same videos. The averaged test performance for the second video is the best with a RMSE of 267 ± 4 sec or 4.4 ± 0.1 min and R^2 of 0.8 ± 0.0. The overall score of all five videos is a RMSE of 509 ± 9 sec or 8.5 ± 0.2 min and R^2 of 0.6 ± 0.0. The fifth video of the test set shows a negative R^2.

TABLE 4.2

PERFORMANCE METRICS RANDOM FOREST REGRESSION MODEL

Videos val set	**RMSE (s)**	**RMSE (min)**	**R^2**
1	445 ± 172	7.4 ± 2.9	0.3 ± 0.8
2	517 ± 250	8.6 ± 4.2	0.7 ± 0.2
3	451 ± 96	7.5 ± 1.6	0.4 ± 0.5
4	480 ± 328	8.0 ± 5.5	0.6 ± 0.5
5	376 ± 156	6.3 ± 2.6	0.6 ± 0.4
overall	454 ± 200	7.6 ± 3.3	0.5 ± 0.5

Videos test set	**RMSE (s)**	**RMSE (min)**	**R^2**
1	400 ± 8	6.7 ± 0.1	0.8 ± 0.0
2	267 ± 4	4.4 ± 0.1	0.9 ± 0.0
3	754 ± 12	12.6 ± 0.2	0.6 ± 0.0
4	476 ± 9	7.9 ± 0.2	0.8 ± 0.0
5	648 ± 14	10.8 ± 0.2	-0.2 ± 0.1
overall	509 ± 9	8.5 ± 0.2	0.6 ± 0.0

Table 4.2: Performance metrics of the five-fold cross-validated RF model with mean and std.

The results of the RF model are visualised in figure 4.6 for the same three videos of the test set, showing the predicted RPT after each surgical phase in relation to the true RPT with a five min acceptance range and the preoperative estimate of 45 min that is used as a standard in the MMC. The prediction is made after the surgical phase has ended. The first video shows a prediction within the five min range for all but the Preparation phase, the second for all phases and third also for all but the Preparation phase. In all videos, the predicted RPT is closer to the true RPT than the 45 min estimate. The high RMSE of video one, shown

in table 4.2, results from the estimate for the Preparation phase. The R^2 is still high as the predictions for all other phases are close to the truth. Also for the third video, the RMSE results from the RPT prediction after the Preparation phase and the R^2 is 0.6 as the other predictions are in the range of the true values of the RPT. For the second video, the predictions are close to the truth except for ClippingTransection. The RMSE results from that prediction, however the R^2 is 0.9.

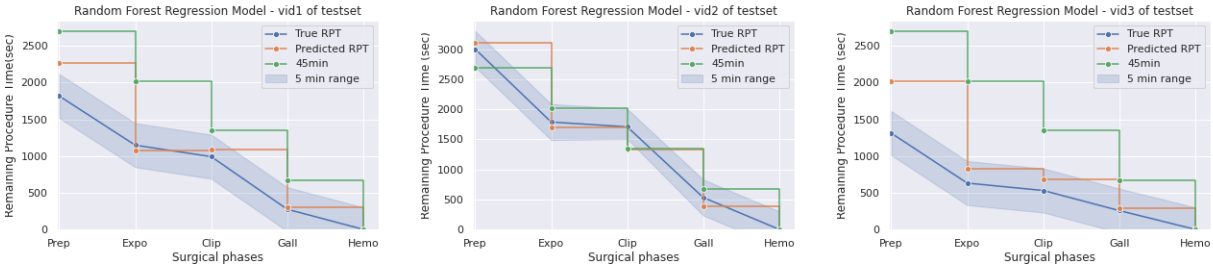

Figure 4.6: Graphical plots for RF model of the true, predicted RPT, five min range and 45 min estimate for the surgical phases.

Support Vector Regression

The performance metrics used to evaluate the SVR model in training and testing are the RMSE and R^2. The validation performance of the model during training is presented on top table 4.3 and for the test set on the bottom. The validation and test results are the averaged results over the five folds of training with cross-validation, so in total of 25 videos. The averaged validation performance for the fifth video is the best with a RMSE of 369 ± 347 sec or 6.1 ± 5.8 min and R^2 of 0.5 ± 0.8. The overall score of all five videos is a RMSE of 557 ± 294 sec or 9.3 ± 4.9 min and R^2 of 0.3 ± 0.9. For the test set, the performance over the five folds is from the five same videos. The averaged test performance for the fourth video is the best with a RMSE of 61 ± 16 sec or 1.0 ± 0.3 min and R^2 of 1.0 ± 0.0. The overall score of all five videos is a RMSE of 709 ± 16 sec or 11.8 ± 0.3 min and R^2 of 0.3 ± 0.1. The second video of the test set shows a negative R^2.

TABLE 4.3

PERFORMANCE METRICS SUPPORT VECTOR REGRESSION MODEL

Videos val set	**RMSE (s)**	**RMSE (min)**	**R^2**
1	570 ± 292	9.5 ± 4.9	0.2 ± 1.0
2	675 ± 331	12.9 ± 4.6	0.2 ± 0.5
3	592 ± 177	9.9 ± 2.9	0.0 ± 1.0
4	477 ± 376	8.0 ± 6.3	0.4 ± 0.9
5	369 ± 347	6.1 ± 5.8	0.5 ± 0.8
overall	557 ± 294	9.3 ± 4.9	0.3 ± 0.9

Videos test set	**RMSE (s)**	**RMSE (min)**	**R^2**
1	370 ± 15	6.2 ± 0.2	0.5 ± 0.0
2	976 ± 20	16.3 ± 0.3	-0.7 ± 0.2
3	992 ± 15	16.5 ± 0.2	-0.3 ± 0.0
4	61 ± 16	1.0 ± 0.3	1.0 ± 0.0
5	845 ± 15	14.1 ± 0.2	0.3 ± 0.0
overall	709 ± 16	11.8 ± 0.3	0.3 ± 0.1

Table 4.3: Performance metrics of the five-fold cross-validated SVR model with mean and std.

The results of the SVR model are visualised in figure 4.7 for the same three videos of the test set. The results show the predicted RPT after each surgical phase in relation to the true RPT with a five min acceptance range and the preoperative estimate of 45 min that is used as a standard in the MMC. The prediction is made after the surgical phase has ended. The first video shows a prediction within the five min range for all the surgical phases, the second only for the last phase and third for not one phase. In the first and third videos, most the predicted RPT is closer to the true RPT than the 45 min estimate. The RMSE of video one, shown in table 4.3, results from the estimate for the HemostasisPackagingRetraction phase. The R^2 is 0.5, as the predictions for all phases are within the acceptance range. The second and third video have a high RMSE as all predictions, except of the HemostasisPackagingRetraction phase of video two, are out of the acceptance range. The R^2 for both videos is negative.

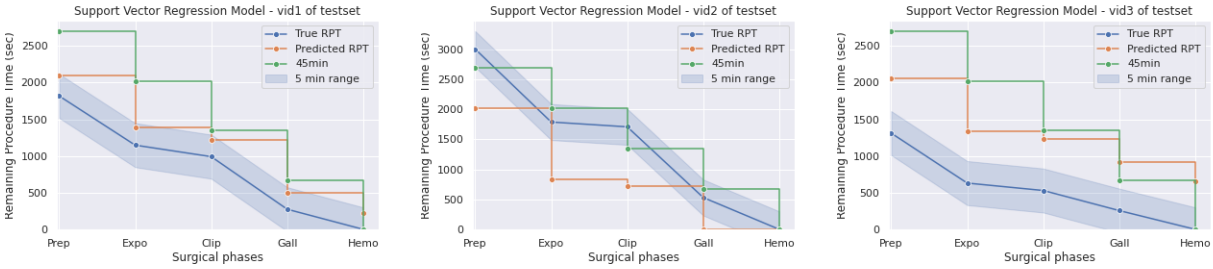

Figure 4.7: Graphical plots for SVR model of the true, predicted RPT, five min range and 45 min estimate for the surgical phases.

Statistical analysis of the models

The statistical significance of the differences between the performance metrics, RMSE and R^2, over the predicted RPTs of the test set by the models, are assessed by a log likelihood function with a CI of 95% and corresponding P-value. The results of the statistical analysis of the regression models are shown in table 4.4. The P-values of the log likelihood show that the difference between the results of the models is not based on coincidence, except for the R^2 of LR and RF.

TABLE 4.4

STATISTICAL ANALYSIS REGRESSION MODELS

Models	**RMSE**	**R^2**
LR - RF	P = 0.02	P = 0.13
LR - SVR	P = 0.03	P = 0.01
RF - SVR	P = 0.02	P = 0.02

Table 4.4: Statistical analysis of the regression model on the RMSE and R^2 performance metrics.

4.5 Discussion

4.5.1 Research question and aim

The aim of this study was to develop an ML-model that can accurately predict the RPT based on the temporal data from LC procedures. The network was solely trained the temporal information of the five surgical phases in the phase duration dataset. To achieve this goal, multiple regression models were used to predict the RPT after each of the five surgical phases.

4.5.2 Discussion of results

Linear Regression

The validation set showed higher overall performance than for the test set, by a reduction RMSE of 47 sec and increase of 0.1 in R^2. The main difference in RMSE is caused by the high RMSE of video three and five and the negative R^2 of video five. The negative R^2 represents that the model predicts worse than taking the mean value, not using information from the variables. The model does not fit to the data for video five. The higher performance on the validation set is achieved, however, with an increase of std in the metrics as there is a higher variation in the input data of the phase durations. The overall RMSE of 10.1 ± 0.6 min shows better performance compared to the MAE of 15.6 min described by Twinanda et al.[75] and MAE of 36.7 min of Bodenstedt et al.[89] These studies were, however, performed on other datasets with different model pipelines. The graphical plots of the first three videos of the test set show that the initial estimate for the RPT after Preparation is the mean LC procedure duration of 2357 sec. For almost all videos, the high RMSE results from this initial estimate. The Preparation phase misses in some videos and has a higher std than mean value. The model does not seem to take in any information of this phase as it always predicts the mean total procedure time. In all videos, the predicted RPT comes closer to the true RPT as the number of input phase durations increases. However, the value of the RPT becomes lower so that results in a higher chance that the predicted value is close to the true value. The LR model outperforms the 45 min estimate in the first and third video. In the second video, the phase durations are higher than the mean values and closer to the 45 min estimate.

Random Forest regression

The validation set showed higher performance overall that for the test set, only by a reduced RMSE of 55 sec. The main difference in RMSE is caused by the high RMSE of video three and five and the negative R^2 of video five. This means that the model predicts worse than taking the mean value and the model does not fit to the data for video five. The higher performance on the validation set comes also, however, with an increase of std in the RMSE and R^2, as there is a higher variation in the input data of the phase durations. The overall RMSE of 8.5 ± 0.2 min shows better performance compared to the MAE of 15.6 min described by Twinanda et al.[75] and MAE of 36.7 min of Bodenstedt et al.[89] These studies were, however, performed on other datasets with different model pipelines. The graphical plots of the first three videos of the test set show that the initial estimate for the RPT after Preparation variates with the input data. In the first and third video, the estimate is far out of the acceptance range which results in a high RMSE from this initial estimate. In the second video, the initial estimate is close to the true value. This finding indicates that the RF model can extract information from the input data of the Preparation phase with high std, but still has a lot of variation. In the first and third video, the predicted RPT comes closer to the true RPT as the number of input phase durations increases. However, the value of the RPT becomes lower so that results in a higher chance that the predicted value is close to the true value. The RF model outperforms the 45 min estimate in all three videos, as for the second video the estimates are quite close.

Support Vector Regression

The validation set showed higher overall performance than for the test set, only by a reduction in RMSE of 152 sec. The main difference in RMSE is caused by the high RMSE of video two, three and five and the negative R^2 of videos two and three. This indicates that the model predicts worse than taking the mean value and the model does not fit to the data for video two and three. The higher performance on the validation set comes also, however, with an increase of std in the RMSE and R^2, as there is a higher variation in the input data of the phase durations. The result of video four is very interesting, as it shows the lowest RMSE of 61sec and highest R^2 of 1.0 of all models for the test and validation videos. However, the high RMSE and negative R^2 of videos two and three are on the complete other side of the performance spectrum. This shows that the SVR is inconsistent in the predictions and that the result of video four is more coincidence that the ability of the model. The overall RMSE of 11.8 ± 0.3 min shows better performance compared to the MAE of 15.6 min described by Twinanda et al.[75] and MAE of 36.7 min of Bodenstedt et al.[89] These studies were, however, performed on other datasets with different model pipelines. In the graphical plots of the first video of the test set shows that the estimate for the RPT after each phase is within the acceptance range but only just. The estimate does not seem to improve when more phase duration data is added. In the second and third video, the predicted RPT is far of the true RPT and again does not come closer to the true RPT as the number of input phase durations increases. The SVR model outperforms the 45 min estimate in the first videos, it performs worse for the second video and on the third video both estimates are off.

Statistical analysis

The statistical analysis by a log likelihood ratio of the best performing regression model, RF, and second best model, LR, shows a statistically significant difference for the RMSE with $P = 0.02$ and insignificant difference for the R^2 with $P = 0.13$ of the test set at a CI of 95%. The R^2 of the first video is 0.8 ± 0.0 for both models. The analysis of these models with the worst performing model, SVR, for the LR a significant difference for the RMSE with $P = 0.03$ and R^2 with $P = 0.01$ at a CI of 95%. The RF shows a significant difference in RMSE and R^2 with $P = 0.02$ at a CI of 95% The statistical analysis of the performance metrics on the prediction of the RPT by the regression models on the five test videos, shows that the RF model has the significantly highest performance of the three models on the RMSE with 509 ± 9 sec or 8.5 ± 0.2 min and for the R^2 with 0.6 ± 0.0 only on the SVR.

4.6 Conclusion

This chapter described the second part of this study, which aimed to development a ML model that can accurately predict the RPT based on the temporal data from LC procedures. It can be concluded that it is possible to predict the RPT, using temporal data extracted from the phase detection in LC videos. The statistically significant best model to predict the RPT is, a RF regression model with an overall RMSE of 8.5 ± 0.2 min and R^2 of 0.6 ± 0.0 on the test set. This performance are improved compared to the results of Twinanda et al.[75] and Bodenstedt et al.[89] However, the validation results showed that all models have a high std when evaluated on more videos. This indicates that these models are prone to variation in the data. A larger dataset could improve and reduce variation of the model performance.

CHAPTER 5

5. General discussion and conclusion

This chapter discusses the clinical relevance, study limitations and recommendations for future research. First, the clinical and scientific relevance of the phase detection and RPT results are elaborated upon. Subsequently, the study limitations of the data, network, model and filtering are discussed. At last, the recommendations are stated for future study and perspective on the application of AI networks and models for the prediction of the RPT based on surgical phase detection.

5.1 Clinical and scientific relevance

Looking at the clinical perspective, some considerations have to be taken into account based on the study results. In order for the phase detection network to have clinical value for implementation, the network must be reliable. The reliability of the network can be expressed in the performance metrics. There is no clear threshold defined for the implementation of AI networks in surgery. A threshold that would probably ensure reliability of the network is over 90% in accuracy on all new video data of the LC procedures. Especially, considering some uncertainty in the classification of the network. The unfiltered results of the ResNet50 are below the required accuracy for clinical implementation. After filtering, the accuracy is often higher than the required percentage. However, these results are too inconsistent for clinical implementation. Further improvements in the network, dataset and learning process, as described in detail in the recommendations, might improve the accuracy of the classification by the network and yield the possibility for clinical implementation. The prediction model for the RPT should yield results within the defined acceptance range of five minutes from the actual time for clinical relevance, based on practical implication. The results of the best performing regression model have an average error rate of 8.5 minutes over all five videos of the test set. Hence, these study results are not within the range for clinical implementation.

The study results can, from a scientific perspective, be considered as interesting and promising for the future. There are many studies conducted for the detection of surgical phases that describe state-of-the-art models that yield high performance. However, the detection of surgical phases alone has no clinical value. The use of the temporal information, provided by the detected phases, for the prediction of the RPT shows promising results for the use in clinical practice. Only a few studies investigated the used of temporal information from phase detection for the prediction of the RPT. This study reports lower error rates for the RPT in comparison to those described in previous research.

Even though the application of AI in healthcare is a fast-growing field because of continuous development and new technical possibilities, some hurdles still need to be taken into account before a wide implementation. The clinical practice in hospitals, including surgery, is still practised by nurses, doctors, and surgeons. The amount of tasks the implementation of AI networks can take over is still minimal.

However, the moment to digitalisation of healthcare is started and it is inevitable that an increasing number of tasks will be performed by AI networks in the future. These tasks include in a wide range of clinical applications and variate in difficulty. Examples are checking and processing of EHR information, automated diagnoses based on visual data and/or certain questionnaires in radiology, automatic tissue, tool and surgical phase recognition for performance evaluation and surgical navigation. The AI networks need to have sufficient performance for implementation, which is not the case for the more complex problems. However, it takes time and trust for doctors and surgeons to accept such technical innovations.[94] The results of this study are not in the range for clinical implementation but they indicate that there is a possibility in the future with further improvements.

5.2 Study limitations

Dataset and preparations

The first part of this research focussed more on the quality of the data than the actual classification network. The public Cholec80 dataset was used and revised, as it is the most widely used LC dataset in scientific research for AI applications. Despite this, the present study is characterised by some flaws in the original annotation and the high variation in the Preparation phase due to delayed recoding. The dataset was sampled at one fps resulting in 184579 frames from 80 videos, which is acceptable for reasonable performance. Although the Preparation phase was 3% of the dataset in the original annotations and 2% in the revised, that phase showed a decreased performance compared to the phases with more frames. In the revised annotation guide, the Preparation phase was incorporated for generalizability on multiple datasets. Even though, the added value for the Cholec80 dataset is minimal as this phase is highly inconsistent.

Other studies showed state-of the-art performance are reached on larger data sets.[95] The assumption is that at least 300 LC videos are needed to provide sufficient data for the phase image and phase duration dataset. New acquisitions of LC data from the MMC was not incorporated in this study, due to time limitations as a result of the time-consuming process of ethical approvals. The revised annotation guide could be used to annotate the newly acquired data. More research is needed to investigate whether adding data from two different sources that use variating surgical tools, would improve or even reduce the network's classification performance. There are also other methods to increase the amount of data without adding new videos to the dataset. The video data had a frequency of 25 fps. Higher sample rates are a very simple option to generate more data but is limit in terms of diversity. There is a trade-off point when a higher sample rate will not introduce new information in the network but only add to the computational burden. The most used sample rate in AI research is one fps but no research has been conducted about the optimal sample rate in these applications. The assumption is that one fps probably is under this trade-off point and 25 fps over.

Data configuration for phase detector

The frames of the videos from the Cholec80 are divided over the train, validation and test set according to the split described by Twinanda et al. and Czempiel et al. being 40, eight and 32 respectively.[30 71] This split is adopted for comparability of the results between studies. The configuration of the data is in this setting 50%, 10% and 40%, as in most research with AI a split of 70%, 10% and 20% is used. The choice of data configuration results in a loss of 20% in training data, which is gained in test data.

The reduced training data could result in earlier overfitting of the network and less generalisable network, which affects the performance on the test set. The large test set would give a good representation on data outside the dataset. However, it might be an underestimation of the potential network performance based on this dataset.

Phase detector network choice

In the field of surgical phase detection of LC procedures, many network structures have been proposed starting with CNNs, hybridisations of CNNs with a LSTM or HMM and eventually TCNs. The development in these network structures also increased the network complexity. Most hybridisations make use of a ResNet50 as convolutional feature extractor because of the ability to incorporate feature information of different levels to the classification due to the skipped connections. As discussed previously, this study did not focus on the development of the best performing network structure for surgical phase detection but more on the data quality and the development of a full pipeline to predict the RPT based on visual data of the LC procedure. Therefore, a base-line network was selected with decent performance and that has been well tested for these applications. The network output of the ResNet50, however, was very noisy which limited the detection of the phase transitions and requested filtering before being useful. The filtering option of the ResNet50 resulted in detectable phase transitions, although the post processing step also resulted in artifacts. Retrospectively, a network structure that has more smooth phase recognition output is preferred. The phase detector can easily be replaced in the proposed pipeline.

Moving window filtering

The post processing step on the raw phase detection output of the ResNet50 was a moving widow. The filter was applied after the network had classified some frames as the widow needed to be filled before filtering could start. The window size was often set at ten or 20 frames and the center frame was altered, which resulted in a time delay of five or ten sec from the current frame that is classified. The output was noisy due to the fact that each image was introduced separately, and consecutive images might contain highly different features although being of the same surgical phase. As these are often individual frames that are misclassified, filtering could be used for smoothening. The filtering performance was highly dependent on the fact that the surrounding frames of the misclassified frames were correct. When more than half of the frames in the window were misclassified, the filter had an adverse effect. The correctly classified frames were converted to the wrong class, resulting in the introduction of false phase blocks. The performance of the network was than negatively affected by the filter and artifacts were introduced as the surgical phases can not occur twice in one procedure. Filtering with a moving window showed inconsistency over the dataset and has a high level of subjectivity. A network structure with more smooth phase output is therefore preferred for more reliable and robust results.

Regression models

The regression models that are evaluated for the prediction of the RPT are selected based on their methods. Simple LR is the most straightforward approach of regression by applying a linear approximation on the data and can therefore be used as a base-line model. LR will perform well on data that is close to the mean of the dataset and poor on deviations from that mean. The RF uses the power of the individual decision trees to make non-linear predictions over the estimator in discrete steps. In general, RF produce better results than LR as they are able to create estimates for missing data. The downside of RF is the inability to

extrapolate outside unseen data. The SVR can use multiple functions and optimisation criterions which enables more possibilities for optimalisation. The used function in this study the third-degree polynomial. The SVR works well with a clear margin of separation between the data and in high dimensional spaces. However, the required training time is substantially higher than the LR and RF. The SVR does not perform well when the dataset contains noise. All three regression models use different methods, each with their own strengths and weaknesses. The RF showed improved performance over the LR and SVR models on the phase duration dataset of the Cholec80. New LC data would require assessment of all three models as the configuration might be more favourable for one of the other models. There is no evidence that these models are the most suited for this application but give an impression of the possibilities. A more extensive study could focus on the evaluation of more regression models.

5.3 Recommendations

Temporal Convolutional Networks

Future research into the application of new state-of-the-art DL networks for phase detection of LC procedures could be conducted to improve the study results and extend to clinical applications. The ResNet50 architecture showed decent performance measured over all frames. The output was, however, quite noisy over the complete procedure. This results from the fact that the frames are introduced individually and no previous information is incorporated in the classification. The phase detection output has to be smooth in order to be able to detect the phase transitions. Research has shown that the incorporation of temporal information for phase detection results in improved performance and more smooth phase output. The TCN incorporates temporal information in addition to the visual features. The first TCN is proposed by Lea et al. in 2016 for video-based action segmentation.[95] The TCN combines the low-level features extracted by a CNN with the high-level temporal information extracted by a RNN in an encoder-decoder architecture. A TCN takes a series of frames of a certain length and uses the information of all these frames for the classification of an individual frame. The classified phase at time *t* is only convolved from the current frame and frames that occurred before *t*, causal convolution. Czempiel et al. showed the use of TCN for surgical workflow recognition on the Cholec80 with an accuracy of 88.56 %.[71]

K-fold cross-validation for phase detector

In the current research, the hyperparameter optimisation is performed by means of a sweep, with the same data configuration as the studies of Twinanda et al. and Czempiel et al.[30 71] The best performing model is selected based on the validation performance metrics. The implementation of K-fold cross-validation would give a more robust analysis into the optimal hyperparameters settings. The current fixed train and validation set configuration, could result in a network that is optimised specific to the characteristics of the videos in the validation set. Through evaluation of the hyperparameter by K-fold cross-validation, the settings could be adjusted based on the performance of more videos and make optimal use of the available data. Eventually, this technique will ensure comprehensive training of the network but comes at a cost of prolonged computational time.

Extension of the LC dataset with data from the MMC

Each year around 600 LC procedures are performed in the MMC. That is a potential source of information that could fast-forward the research in surgical phase detection and the application for predicting the RPT. The ethics board has given its approval to use this information for scientific research. The inclusion pipeline is set up and acquisition of new patients is slowly starting. After including a sufficient amount of patients, the video data has to be annotated. For this study, fully manual annotation was required as no network was trained on the revised phase annotations. An assumption is that the trained network on the Cholec80 data could be used to classify the frames of the MMC data. These classifications have to be examined and corrected were needed manually. This process works faster than the fully manual annotation approach and makes use of the gained knowledge of the network. The network could also be trained on the already examined data for improved performance and reduces the corrections.

Balancing the dataset

The Cholec80 dataset is imbalanced for both the original and revised annotations. The minority phases contain between 2 - 5% of the frames each. The low amount of frames in the Preparation phase resulted from delayed recording. In the new acquisitions of LC data from the MMC, a selection could be made based on the presence of all phases. This would increase the number of frames in the Preparation phase by some amount. The under performance of the minority phases is most clearly shown in the original annotations. Even though class-weighting was applied, these phases showed a difference of 20% in precision and recall. Another option to counter action on the imbalance of these phases in the procedures is to make the dataset more balanced. There are two options, one is oversampling the data of the minority phases. The used sample rates for the conversion from video to frame data could be inverse proportional to the length of the phase. A downside of this technique is that with high sample rates, the addition of new frames will not introduce new information into the dataset. The consecutive frames of at high sample rates show high similarity. The other technique is that by generating the new dataset from the acquisition of the MMC, a fixed amount for the majority classes is chosen and after that number of patients only new data for the minority classes will be introduced to the dataset. This would reduce the difference between the phases but is limited by the chosen amount and total number of included patients. These techniques should only be applied on the train data, as the validation and test data should resemble the configuration of clinical practice.

Continues RPT predictions

The RPT predictions by the regression models are made based on the phase durations. The durations can be detected after the phase has passed. During the LC procedure the model makes five predictions. No time updates are given between these estimates. The incorporation of the past between the detected phase transitions generates a continuous input of temporal information. RPT can continuously be adjusted by subtracting the past time from the estimates made after the phase transition.. Although the RPT is continuously displayed, they are still based on the same phase duration information as the five separate estimates. It might be more intuitive to receive a constant update about the expected time of arrival (ETA) of the procedure. However, the estimate has the same error but might give a false impression of being more accurate.

Statistical analysis for phase difference

In this study a simple correction for the difference in guess chance is used to give an indication in the performance difference between the two annotations. In order to prove the statistical significance of the performance difference, more advanced statistical analysis should be applied. A Monte-Carlo simulation would be a suited method. This method is a computerized mathematical technique that provides a range of possible outcomes. For both networks individually, thousands of simulations should to be conducted. The simulation results will give a mean value and std for the performance metrics of both networks. The mean value and std of both networks can be compared by an ANOVA test to evaluate the statistical significance of the difference between the results with a defined CI.

5.4 Conclusion

This study aimed to develop two AI algorithms for the automatic analysis of laparoscopic video data and prediction of the RPT. The DL network classified the frames from intraoperative laparoscopic videos in the surgical phases of the LC procedure. It can be concluded that the phase classifications showed decent performance for a base-line network. Post processing of the phase output removed the noisy character but was susceptible to artifacts. TCNs are advised for future research. This study additionally aimed to investigate the importance of adequate labelling for detecting surgical phases of the LC procedure. The performance metrics indicated that the revised annotations improved 6.0%, 5.8%, 6.2% and 6.0%, for accuracy, precision, recall and F1-score respectively. The ML model accurately predicted the RPT based on the phase durations of the LC procedure. The RF regression model showed to be the best model to predict the RPT, with an overall RMSE of 8.5 min and R^2 of 0.6 on the test set. Hereby, this research model improves on the performance stated by Twinanda et al.[75] and Bodenstedt et al.[89] The RPT prediction model did, however, not yield results that are within the standards for use in clinical practice. Further improvements on the network, dataset and learning process, as described in the recommendations, might enable the possibility for clinical implementation. The added value in clinical practice for patients, surgeons and OR staff is more optimal OR planning. Which may reduce delays or even cancellation of subsequent procedures, resulting in shorter waiting times for patients and less overtime for OR personnel.

References

1. Centraal Bureau voor de Statistiek, “Operaties in het ziekenhuis: soort opname, leeftijd en geslacht. 1995-2010” (2014). https://opendata.cbs.nl/statline/#/CBS/nl/dataset/80386NED/table?fromstatweb

2. Hassler, K. R and Jones, M. W. (2017). Laparoscopic Cholecystectomy. StatPearls. StatPearls Publishing. Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK448145/

3. H. M. Atta, A. A. Mohamed, A. M. Sewefy, A.-F. S. Abdel-Fatah, M. M. Mohammed, and A. M. Atiya, “Difficult Laparoscopic Cholecystectomy and Trainees: Predictors and Results in an Academic Teaching Hospital,” 2017, doi: 10.1155/2017/6467814.

4. Lap cholecystectomy Tamil Nadu | Gall bladder Removal India. (n.d.). Retrieved March 4, 2021, from https://www.guruhospitals.com/lap-cholecystectomy

5. J. Lange and G. Kleinrensink, \The gallbladder and bile ducts," in Surgical Anatomy of the Abdomen, 1st ed. Elsevier, 2002, ch. 10, p. 274.

6. S. Virupaksha, “Consequences of spilt gallstones during laparoscopic cholecystectomy," The Indian journal of surgery, vol. 76, no. 2, pp. 95-99, 2014. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/24891771https://www.ncbi.nlm.nih.gov/pubmed/ 24891771

7. A. Nooghabi, M. Hassanpour, and A. Jangjoo, “Consequences of Lost Gallstones during Laparoscopic Cholecystectomy: A Review Article," Surgical Laparoscopy, Endoscopy and Percutaneous Techniques, vol. 26, no. 3, pp. 183-192, 2016. [Online]. Available: www.surgical-laparoscopy.comwww.surgical-laparoscopy.com

8. A. van Dijk, M. van der Hoek, M. Rutgers, P. van Duijvendijk, S. Donkervoort, P. de Reuver, and M. Boermeester, “Efficacy of Antibiotic Agents after Spill of Bile and Gallstones during Laparoscopic Cholecystectomy," Surgical Infections, vol. 20, no. 4, pp. 298-304, 2019.

9. J. Zehetner, A. Shamiyeh, and W. Wayand, “Lost gallstones in laparoscopic cholecystectomy: all possible complications," American Journal of Surgery, vol. 193, no. 1, pp. 73-78, 1 2007.

10. Haribhakti, S. P., & Mistry, J. H. (2015, April 1). Techniques of laparoscopic cholecystectomy: Nomenclature and selection. Journal of Minimal Access Surgery. Medknow Publications. https://doi.org/10.4103/0972-9941.140220

11. Gallbladder Surgery Single-Incision - Georgia SurgiCare. (n.d.). Retrieved March 5, 2021, from https://www.georgiasurgicare.com/same-day-surgery-center/single-incision-gallbladder-surgery/

12. Marchi, D., Esposito, M. G., Gentile, I. G., & Gilio, F. (2014). Laparoscopic Cholecystectomy: Training, Learning Curve, and Definition of Expert. Laparoscopic Cholecystectomy, 141–147. doi:10.1007/978-3-319-05407-0_11

13. Moore MJ, Bennett CL (1995) The learning curve for laparoscopic cholecystectomy. The Southern Surgeon’s Club. Am J Surg 170:55–59

14. Giger, U. F., Michel, J. M., Opitz, I., Inderbitzin, D. T., Kocher, T., & Krähenbühl, L. (2006). Risk Factors for Perioperative Complications in Patients Undergoing Laparoscopic Cholecystectomy: Analysis of 22,953 Consecutive Cases from the Swiss Association of Laparoscopic and Thoracoscopic Surgery Database. Journal of the American College of Surgeons, 203, 723–728. https://doi.org/10.1016/j.jamcollsurg.2006.07.018

15. Schijven M, Jakimowicz J (2003) Construct validity – experts and novices performing on the Xitact LS500 laparoscopy simulator. Surg Endosc 17:803–810

16. Cardoen, B., Demeulemeester, E., Beliën, J.: Operating room planning and scheduling: A literature review. European Journal of Operational Research 201(3), 921–932 (2010)

17. Eijkemans MJC, Van Houdenhoven M, Nguyen T et al (2010). Predicting the unpredictable: a new prediction model for operating room times using individual characteristics and the surgeon’s estimate. Anesthesiology. https ://doi.org/10.1097/ALN.0b013e3181c294c 2

18. Dexter F, Ph D, Epstein RH et al (2017). Making management decisions on the day of surgery based on operating room efficiency and patient waiting. J Am Soc Anesthesiol 101:1444–1453

19. Edelman ER, Van KSMJ, Hamaekers AEW et al (2017). Improving the prediction of total surgical procedure time using linear regression modeling. Front Med 4:1–5. https ://doi.org/10.3389/fmed.2017.00085

25. van Eijk RPA, Van V-B, Kazemier G, Eijkemans MJC (2016). Effect of individual surgeons and anesthesiologists on operating room time. Anesth Anal. https ://doi.org/10.1213/ANE.0000000000 001430

20. Gupta N, Ranjan G, Arora MP et al (2013) Validation of a scoring system to predict difficult laparoscopic cholecystectomy. Int J Surg 11:1002–1006. https ://doi.org/10.1016/j.ijsu.2013.05.037

21. Guédon, A. C. P., Meij, S. E. P., Osman, K. N. M. M. H., & Kloosterman, H. A. (2020). Deep learning for surgical phase recognition using endoscopic videos. Surgical Endoscopy, (0123456789). https://doi.org/10.1007/s00464-020-08110-5

22. Kayis, E., Wang, H., Patel, M., Gonzalez, T., Jain, S., Ramamurthi, R., Santos, C., Singhal, S., Suermondt, J., Sylvester, K.: Improving Prediction of Surgery Duration using Operational and Temporal Factors. In: AMIA Annu. Symp. Proc., pp. 456–462 (2012)

23. Wiegmann DA, ElBardissi AW, Dearani JA et al (2007) Disruptions in surgical flow and their relationship to surgical errors: an exploratory investigation. Surgery 142:658–665. https://doi.org/10.1016/j.surg.2007.07.034

24. Arora S, Hull L, Sevdalis N et al (2010) Factors compromising safety in surgery: stressful events in the operating room. Am J Surg 199:60–65. https ://doi.org/10.1016/j.amjsu rg.2009.07.036

25. Blum T, Padoy N, Feußner H, Navab N (2008) Modeling and online recognition of surgical phases using hidden Markov models. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) LNCS 5242:627–635. https://doi.org/10.1007/978-3-540-85990 -1-75

26. Guédon ACP, Paalvast M, Meeuwsen FC et al (2016) ‘It is Time to Prepare the Next patient’ Real-Time Prediction of Procedure Duration in Laparoscopic Cholecystectomies. J Med Syst. https://doi.org/10.1007/s1091 6-016-0631-1

27. Meeuwsen FC, van Luyn F, Blikkendaal MD et al (2019) Surgical phase modelling in minimal invasive surgery. Surg Endosc. https://doi.org/10.1007/s0046 4-018-6417-4

28. Akilambigai, A., & Vijayashanthi, K. (2018). An Overview of Clinical Applications of Artificial Intelligence. International Journal of Scientific Research in Computer Science, Engineering and Information Technology 2018 IJSRCSEIT, 3(4), 2456–3307. Retrieved from http://ijsrcseit.com/paper/CSEIT184302.pdf

29. Abbing, J. R. (2020). Semantic segementation of minimally invasis anti-reflux surgery video using U-NET Machine Learning. Faculty of Science and Technology

30. Twinanda, A. P., Shehata, S., Mutter, D., Marescaux, J., Mathelin, M. De, & Padoy, N. (2016). EndoNet : A Deep Architecture for Recognition Tasks on Laparoscopic Videos, (February). https://doi.org/10.1109/TMI.2016.2593957

31. S. Bodenstedt, D. Rivoir, A. Jenke, M. Wagner, M. Breucha, B. Müller-Stich, S.T. Mees, J. Weitz, and S. Speidel. (2019) Active learning using deep Bayesian networks for surgical workflow analysis. International Journal of Computer Assisted Radiology and Surgery, 14(6):1079–1087,

32. Z. Wang and A. Majewicz Fey, (2018) Deep learning with convolutional neural network for objective skill evaluation in robot-assisted surgery, International Journal of Computer Assisted Radiology and Surgery, vol. 13, no. 12, pp. 1959-1970.

33. Figueroa, R. L. Zeng-Treitler, Q. Kandula, S. & Ngo, L. H. (2012) Predicting sample size required for classification performance. BMC Med. Inform. Decis. Mak. 12(8).

34. Beleites, C. Neugebauer, U. Bocklitz, T. Krafft, C. & Popp, J. (2013) Sample size planning for classification models. Analytica Chimica Acta, 760(14): 25-33.

35. Hong, S., Lee, J., Park, B., Choi, M.-K., Jin Hyung, W., Alwusaibie, A. A., … Park, S. (2021). Rethinking Generalization Performance of Surgical Phase Recognition with Expert-Generated Annotations.

36. Gerkema, M. H., Broeders, I. A. M. J., & Heijden, F. Van Der. (2020). Thesis Technical Medicine Deep learning for identification of gallbladder leakage during laparoscopic cholecystectomy.

37. N. Padoy, (2019) Machine and deep learning for workflow recognition during surgery, Minimally Invasive Therapy and Allied Technologies, vol. 28, no. 2, pp. 82-90.

38. A. Twinanda, G. Yengera, D. Mutter, J. Marescaux, and N. Padoy, (2018) RSDNet: Learning to Predict Remaining Surgery Duration from Laparoscopic Videos Without Manual Annotations, IEEE Transactions on Medical Imaging, vol. 38, no. 4, pp. 1069-1078.

39. Chatterjee, C. C. (2019). Basics of the Classic CNN. Retrieved from https://towardsdatascience.com/basics-of-the-classic-cnn-a3dce1225add

40. Networks, R. (2021). Role of Bias in Neural Networks Retrieved from https://intellipaat.com/community/253/role-of-bias-in-neural-networks

41. D. Liu, (2017) A Practical Guide to ReLU, Retrieved from https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7

42. Sharma, S. (2017). Activation Functions in Neural Networks. Retrieved from https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

43. Tayal, R. (2020). Deep Learning for Computer Vision. Retrieved from https://towardsdatascience.com/deep-learning-for-computer-vision-c4e5f191c522

44. Stewart, M. (2019). Simple Guide to Hyperparameter Tuning in Neural Networks. Retrieved from https://towardsdatascience.com/simple-guide-to-hyperparameter-tuning-in-neural-networks3fe03dad8594

45. Stewart, M. (2019). Neural Network Optimization. Retrieved from https://towardsdatascience.com/neural-network-optimization-7ca72d4db3e0

46. Brownlee, J. (2019). A Gentle Introduction to Information Entropy. Retrieved from https://machinelearningmastery.com/what-is-information-entropy/

47. Brownlee, J. (2019). A Gentle Introduction to Cross-Entropy for Machine Learning. Retrieved from https://machinelearningmastery.com/cross-entropy-for-machine-learning/

48. Brownlee, J. (2019). Loss and Loss Functions for Training Deep Learning Neural Networks. Retrieved from https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/

49. Koech, K. E. (2020). Cross-Entropy Loss Function. Retrieved from https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e

50. Shrivastava, I. (2020) Handling Class Imbalance by Introducing Sample Weighting in the Loss Function. Retrieved from https://medium.com/gumgum-tech/handling-class-imbalance-by-introducing-sample-weighting-in-the-loss-function-3bdebd8203b4

51. PyTorch 1.9.0, CrossEntropyLoss Documentation. (2021) Retrieved from https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

52. Wudaru. G. (2020). Tips for handling Class Imbalance Problem. Retrieved from https://medium.com/ml-course-microsoft-udacity/tips-for-handling-class-imbalance-problem-fb77c192898e

53. Trehan, D. (2020). Gadient Desent Explained. Retrieved from https://towardsdatascience.com/gradient-descent-explained9b953fc0d2c

54. Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 1–15.

55. Sanghvirajit (2020). A Complete Guide to Adam and RMSprop Optimizer. Retrieved from https://medium.com/analytics-vidhya/a-complete-guide-to-adam-and-rmsprop-optimizer-75f4502d83be

56. Kaur, H. (2020) Understanding Linear Regression Model. Retrieved from https://medium.com/@harmeetkaur.trainer/understanding-linear-regression-model-3c9bfd3e0c34

57. Dave, P. (2020) Linear Regression. Retrieved from https://medium.com/swlh/linear-regression-models-dc81a955bd39

58. Koehrsen, W. (2017) Random Forest Simple Explanation. Retrieved from https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d

59. Ai, Z. (2020) Decision Trees Explained. Retrieved from https://towardsdatascience.com/decision-trees-explained-3ec41632ceb6

60. Sharp, T. (2020) An Introduction to Support Vector Regression (SVR). Retrieved from https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2

61. Aditya, P. (2018) L1 and L2 Regularization. Retrieved from https://medium.com/@aditya97p/l1-and-l2-regularization-237438a9caa6.

62. Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into Imaging. doi:10.1007/s13244-018-0639-9

63. He, Y., & Zhao, J. (2019). Temporal Convolutional Networks for Anomaly Detection in Time Series. In Journal of Physics: Conference Series (Vol. 1213). Institute of Physics Publishing. https://doi.org/10.1088/1742-6596/1213/4/042050

64. Tamer Abdulbaki Alshirbaji, Jalal, N. A., & Möller, K. (2020). A convolutional neural network with a two-stage LSTM model for tool presence detection in laparoscopic videos. Current Directions in Biomedical Engineering, 6(1), 1–4. https://doi.org/10.1515/cdbme-2020-0002

65. Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE,1989,77,257-286

66. Stauder, R., Ostler, D., Kranzfelder, M., Koller, S., Feußner, H., & Navab, N. (2016). The TUM LapChole dataset for the M2CAI 2016 workflow challenge.

67. Gagana, B. (2019). Class Activation Maps. https://medium.com/@GaganaB/class-activation-maps-551477720679

68. Namazi, B., Sankaranarayanan, G., & Devarajan, V. (2021). A contextual detector of surgical tools in laparoscopic videos using deep learning. Surgical Endoscopy. doi:10.1007/s00464-021-08336-x

69. Gagniuc, Paul A. (2017). Markov Chains: From Theory to Implementation and Experimentation. USA, NJ: John Wiley & Sons. pp. 1–256. ISBN 978-1-119-38755-8.

70. Yengera, G., Mutter, D., Marescaux, J., & Padoy, N. (2018). Less is more: Surgical phase recognition with less annotations through self-supervised pre-training of CNN-LSTM networks. ArXiv.

71. Czempiel, T., Paschali, M., Keicher, M., Simson, W., Feussner, H., Kim, S. T., & Navab, N. (2020). TeCNO: Surgical phase recognition with multi-stage temporal convolutional networks. ArXiv, 1–10.

72. Den Boer, K. T., Dankelman, J., Gouma, D. J., & Stassen, H. G. (2001). Peroperative analysis of the surgical procedure. Surgical Endoscopy And Other Interventional Techniques, 16(3), 492–499. doi:10.1007/s00464-001-8216-5

73. F. Milletari (2021) VisionWorks [Source code]. https://github.com/faustomilletari/VisionWorks.

74. He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). DOI: 10.1109/CVPR.2016.90

75. Aksamentov, I., Twinanda, A. P., Mutter, D., Marescaux, J., & Padoy, N. (2017). Deep Neural Networks Predict Remaining Surgery Duration from Cholecystectomy Videos. Medical Image Computing and Computer-Assisted Intervention − MICCAI 2017, 586–593. doi:10.1007/978-3-319-66185-8_66

76. Wang, J., Cabrera, J., Tsui, K.-L., Guo, H., Bakker, M., & Kostis, J. B. (n.d.). Predicting Surgery Duration from a New Perspective: Evaluation from a Database on Thoracic Surgery.

77. ShahabiKargar, Z., Khanna, S., Good, N., Sattar, A., Lind, J., & O'Dwyer, J. (2014). Predicting Procedure Duration to Improve Scheduling of Elective Surgery. PRICAI 2014: Trends in Artificial Intelligence, 998–1009. doi:10.1007/978-3-319-13560-1_86

78. Shahabikargar, Z., Khanna, S., Sattar, A., & Lind, J. (2017). Improved Prediction of Procedure Duration for Elective Surgery. https://doi.org/10.3233/978-1-61499-783-2-133

79. Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V… (2021). Cross-validation: evaluating estimator performance. https://scikit-learn.org/stable/modules/cross_validation.html

80. S. Kaul. (2020) Deeply Explained Cross-Validation in ML/AI. https://medium.com/analytics-vidhya/deeply-explained-cross-validation-in-ml-ai-2e846a83f6ed

81. Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geoscientific Model Development, 7(3), 1247–1250. doi:10.5194/gmd-7-1247-2014

82. De Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2016). Mean Absolute Percentage Error for regression models. Neurocomputing, 192, 38–48. doi:10.1016/j.neucom.2015.12.114

83. Chicco, Davide; Warrens, Matthijs J.; Jurman, Giuseppe (2021). "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation". PeerJ Computer Science. 7 (e623): 1–24. doi:10.7717/peerj-cs.623.

84. Ritter, A.; Muñoz-Carpena, R. (2013). "Performance evaluation of hydrological models: statistical significance for reducing subjectivity in goodness-of-fit assessments". Journal of Hydrology. 480 (1): 33–45. Bibcode:2013JHyd..480...33R. doi:10.1016/j.jhydrol.2012.12.004.

85. Macario, A., Dexter, F.: (1999) Estimating the duration of a case when the surgeon has not recently scheduled the procedure at the surgical suite. Anesth. Analg. 89, 1241–1245

86. Kayis, E., Wang, H., Patel, M., Gonzalez, T., Jain, S., Ramamurthi, R.J., Santos, C.A., Singhal, S., Suermondt, J., Sylvester, K.: (2012) Improving prediction of surgery duration using operational and temporal factors. In: AMIA

87. Dexter, F., Epstein, R.H., Lee, J.D., Ledolter, J.: (2009) Automatic updating of times remaining in surgical cases using bayesian analysis of historical case duration data and instant messaging updates from anesthesia providers. Anesth. Analg. 108(3), 929–940

88. Guédon, A.C.P., Paalvast, M., Meeuwsen, F.C., Tax, D.M.J., van Dijke, A.P., Wauben, L., van der Elst, M., Dankelman, J., van den Dobbelsteen, J. (2015) Real-time estimation of surgical procedure duration. In: International Conference on E-health Networking, Application & Services, pp. 6–10

89. Bodenstedt S, Wagner M, Mündermann L, Kenngott H, Müller-Stich B, Breucha M, Mees ST, Weitz J, Speidel S (2019) Prediction of laparoscopic procedure duration using unlabeled, multimodal sensor data. Int J Comput Assist Radiol Surg 14:1089–1095

90. A. Gramfort, F. Pedregosa, O. Grisel, V. Michel, P. Prettenhofer, M. Blondel, L. Buitinck, M. Morel, G. Patrini, M. Telenczuk. (2021). LinearRegression [Source code]. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

91. G. Louppe, B. Holt, J. Arnaud, F. Hedayati. (2021). RandomForestRegressor [Source code]. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

92. G. Lemaitre, J. du Boisberranger, O. Grisel. (2021). SupportVectorRegression [Source code]. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

93. Robinson, E. (2016). Introduction to Likelihood Statistics. Retrieved April 16, 2021 from: https://harvard.edu/AstroStat/aas227_2016/lecture1_Robinson.pdf

94. van de Kar, N.E., Broeders, I. A. M. J., & Heijden, F. Van Der. (2020). Automatic phase recognition during laparoscopic cholecystectomies with a convolutional and hybrid neural network-based deep learning algorithm.

95. Zhang, B., Abbing, J., Ghanem, A., Fer, D., Barker, J., Abukhalil, R., … Milletarì, F. (2021). Towards accurate surgical workflow recognition with convolutional networks and transformers. Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization, 00(00), 1–8. https://doi.org/10.1080/21681163.2021.2002191

96. Lea, C., Vidal, R., Reiter, A., & Hager, G. D. (2016). Temporal convolutional networks: A unified approach to action segmentation. In European Conference on Computer Vision (pp. 47-54). Springer, Cham