Video analysis for augmented cataract surgery

Hassan Al Hajj

UNIVERSITE BIOLOGIE BRETAGNE SANTE LOIRE

UBO
Université de Bretagne Occidentale

THESE DE DOCTORAT DE

L'UNIVERSITE
DE BRETAGNE OCCIDENTALE
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 605
Biologie Santé
Spécialité : Analyse et traitement de l'information et des images médicales

Par
Hassan ALHAJJ

Video analysis for augmented cataract surgery

Thèse présentée et soutenue à Brest, le 13/07/2018
Unité de recherche : UMR1101 Inserm, LaTIM

Rapporteurs avant soutenance :

Vincent SOLER	Professeur d'Université – Praticien Hospitalier, CHU de Toulouse
Nicolas PADOY	Maitre de conférences, HDR, à l'Université de Strasbourg

Composition du Jury :

Vincent SOLER	Professeur d'Université – Praticien Hospitalier, CHU de Toulouse Président
Nicolas PADOY	Maitre de conférences, HDR, à l'Université de Strasbourg
Gwenolé QUELLEC	Chargé de recherche, HDR, à l'Inserm
Béatrice COCHENER	Professeur d'Université – Praticien Hospitalier, CHRU de Brest Directrice de thèse
Invité(s) Mathieu LAMARD	Ingénieur de recherche à l'Université de Bretagne Occidentale

"The single greatest cause of happiness is gratitude."

Auliq-Ice

Acknowledgements

First and foremost, I would like to thank Almighty Allah SWT for giving me the strength, knowledge, ability and opportunity to undertake this research study and to complete it. Without his blessings, this achievement would not have been possible.

I would like to express my appreciation to Prof. **Béatrice Cochener** for giving me the opportunity to do a Phd. She have given me the opportunity to access a large amount of surgical data and precious advices in order to accomplish the job.

I take pride in acknowledging the insightful guidance of Dr. **Gwenolé Quellec**, my thesis supervisor. He has given me invaluable guidance, inspiration and suggestions in my quest for knowledge. Without his guidance and patience, this thesis would not have been possible and I shall eternally be grateful to him for his assistance. I am also grateful to DR. **Mathieu Lamard** for his support, encouragement and ideas that have been great contributors in the completion of the thesis. In addition, it would be inappropriate if I omit to mention the name of Dr. **Guy Cazuguel** who has supervised me during my first year of thesis.

I would also like to express my gratitude to my committee members, Prof. **Vincent Soler** and Dr. **Nicolas Padoy**, who examined my thesis work and provided insightful suggestions.

My acknowledgement would be incomplete without thanking the biggest source of my strength and moral support, my family, especially my mother, my father, my brother and my sisters.

Ultimately, I have great pleasure in acknowledging my gratitude to my colleagues and my friends for supporting me during the 3.5 years of my thesis.

Abstract

The digital era is increasingly changing the world due to the sheer volume of data produced every day. In particular, the medical domain is highly affected by this revolution, because analysing this data can be a source of education/support for the clinicians. In this thesis, we propose to reuse the surgery videos recorded in the operating rooms for computer-assisted surgery system. We are chiefly interested in recognizing at each instant of the surgery the surgical gesture being performed in order to provide relevant information. To achieve this goal, this thesis addresses the surgical tool recognition problem, with applications in cataract surgery. The main objective of this thesis is to address the surgical tool recognition problem in cataract surgery videos. In the surgical field, those tools are partially visible in videos and highly similar to one another. To address the visual challenges in the cataract surgical field, we propose to add an additional camera filming the surgical tray. Our goal is to detect the tool presence in the two complementary types of videos: tool-tissue interaction and surgical tray videos. The former records the patient's eye and the latter records the surgical tray activities.

Two tasks are proposed to perform the task on the surgical tray videos: tools changes detection and tool presence detection. First, we establish a similar pipeline for both tasks. It is based on typical classification methods on top of visual learning features. It yields satisfactory results for the tools changes task, however, it badly performs the surgical tools presence task on the tray. Second, we design deep learning architectures for the surgical tool detection on both video types in order to address the difficulties in manually designing the visual features. To alleviate the inherent challenges on the surgical tray videos, we propose to generate simulated surgical tray scenes dataset along with a patch-based convolutional neural network (CNN). Ultimately, we study the temporal information using RNN based on the CNNs results. Contrary to our primary hypothesis, the experimental results shows deficient results for the surgical tool presence on the tray but very good results on the tool-tissue interaction videos. We achieve even better results in the surgical field after fusing the tools changes information from coming the tray and tool presence signals on the tool-tissue interaction videos.

Contents

List of Figures

List of Tables

"The process of scientific discovery is, in effect, a continual flight from wonder"
Albert Einstein

1 Introduction

Chapter Content

We are living in an era where technology is increasingly changing the shape of our world. It is no wonder that the technological advances have highly affected the medical field. They have been translated into medical innovations ranging from medical image acquisition systems to robotic surgical systems. These systems produce a massive data storage relatively unexplored and difficult to be explored manually by the clinicians. Thus, the need for automating the process of extracting information from the medical data. In this context, various computer-aided decision systems have been explored during the last few decades, such as computer-aided diagnosis and computer-assisted surgery. Computer-aided diagnosis can be defined as a diagnosis made by the clinicians who take into account the results of an automated medical opinion. This second opinion is based on finding commonalities between the medical case being studied and the previous ones already diagnosed. This implicitly improves the medical knowledge of the clinicians and supports them in their decision-making task. In addition to the medical diagnostic aid, the medical archived records can also be exploited to provide surgical guidance. Computer-assisted surgery is a field where a surgery is supported by a computer-based tools and methodologies. Various kind of computer-assisted surgical systems can be put in place in the operating room (OR) to guide the clinicians ranging from simply knowing the state of the OR to letting a surgical robot perform some tasks of the surgical procedure. This can be done by analyzing the various signals coming from the equipments installed in the OR. In other words, it is about exploiting these signals in order to tell what is taking place during the surgery. In our team, we are primarily interested in the visual signals captured in the OR. Precisely, we are addressing the surgical videos

recorded during the cataract surgery towards the target of generating warnings and recommendations to the surgeons along the surgery.

Indeed, it is necessary to be able along the line to tell at each instant of the surgery which surgical activity is being done by the surgeons. Numerous studies in the team have been initiated to automatically recognize the surgical activity/step/phase. The obtained results are encouraging when using the surgical tools signals ground truth as an input to the system. In fact, these results highlighted the challenge of detecting the surgical tools in the tool-tissue interaction videos. Yet, these tools have a large variety of shapes and we can only recognize the tools edges in the tool-tissue interaction videos, usually recorded by an endoscope or a microscope. To overcome these challenges, in this thesis, we propose to extend the studied field of view from only the operative field to both the operative field and the surgical tray by recording a second video stream filming the surgical tray. In contrast to the tool-tissue interaction field of view, the surgical tools are more easily recognizable on the surgical tray. Nevertheless, it is still challenging to detect the surgical tools on the tray due to its specification: it holds too many objects other than the tools used in the operative field and the surgeons accomplish some preliminary actions before using the tools inside the surgical field. In this thesis, we propose to jointly analyze the tool-tissue interaction video and the surgical tray video in order to exploit their associated advantages: (1) the details of the tools edges and how they are used in the surgical field. (2) the recognizable version of the surgical tools with their installation environment on the surgical tray.

With an estimated nineteen million operations performed annually, cataract surgery is the most common surgical procedure. It is considered as an ideal field of study with potential applications in real-time decision support, report generations and surgical training. Due to the limited number of public surgical tool datasets, our first contribution is the generation of a large dataset for surgical tool recognition in the cataract surgery. We have collected and labelled tens of cataract surgery videos using a web application built specifically for this task. They are containing real surgical procedures where each surgery is recorded in two videos: tool-tissue interaction video and surgical tray video. They are the result of a close collaboration with the ophthalmology department of Brest University Hospital. This dataset permits the evaluation of the proposed approaches in this thesis. In addition, we released publicly, in 2017, the tool-tissue interaction videos in the context of a challenge called CATARACTS[1] in order to detect the surgical tool presence in the surgical field videos. In 2018, we released the surgical tray videos along with the tool-tissue interaction videos in the context of EndoVis/CATARACTS[2], a sub-challenge of MICCAI EndoVis challenge for the sake of pushing forward the results obtained in the first challenge by providing a new technical challenge represented by the tray videos.

As second contribution, two different surgical tool recognition pipelines are proposed in this thesis. One is a patch-based approach using traditional classification methods on top of handcrafted features or learning features. Yet, designing discriminative features is not a trivial task due to the challenges in both types of videos. We propose to automatically learn the visual features in both types of videos us-

[1] https://cataracts.grand-challenge.org/

[2] https://cataracts2018.grand-challenge.org

ing deep learning methods. Moreover, we propose to generate simulated surgical tray datasets, expectedly sidestepping the inherent challenges of the surgical tray. Therefore, the second approach is based on well-known convolutional neural networks (CNN) architectures on top of these simulated datasets for the surgical tray videos. These CNNs are used as well on the tool-tissue interaction videos. In addition, the temporal constraints are deemed as a significant component for any surgical activity recognition system. Getting leverage of the temporal constraints in a surgical tool recognition system can have as well a noticeable effect on the performance of the system. Then, to incorporate the temporal information, Long-Short Term Memory (LSTM) network architecture is used on top of the visual features extracted for both videos.

1.1 Outline

In this thesis, we discuss only the work where we are the main contributor, however, in Appendix C, we present the papers issued from this thesis where we are a secondary contributor.

Chapter 2 introduces the context of this thesis and includes a literature review of the methods proposed in the surgical activity recognition and the surgical tool recognition fields.

Chapter 3 describes the cataract surgery, the dataset collection and the challenges present in it.

Chapter 4 presents several surgical tool recognition pipelines based on traditional classification and feature extraction approaches on the surgical tray videos.

Chapter 5 describes the deep learning based solution to address the surgical tool presence detection in the tool-tissue interaction and the surgical tray videos. It explores as well the simulated surgical tray dataset.

Chapter 6 contains the fusion of the surgical tool information coming from both types of videos, expectedly boosting the performance of the system.

Chapter 7 concludes the work done in this thesis and introduces several possible improvements of the methodologies and directions for future work.

"If you want the present to be different from the past, study the past."

Baruch Spinoza

2

Context and Literature Review

Chapter Content

Along with the tremendous technological progress during the last few decades, the digital data became increasingly involved in all aspect of our life. From tiny mass storage in the sixties to today's Data Centers, the humanity has reached a historic milestone. In addition to the use of computers for scientific computing, the use of "Big Data" has been added to extract knowledge and expertise directly from the digital data archived in all areas: social networks, economics, finance, ecology, cartography, multimedia, etc. With the "Big Data" emanating from various digital sources, its importance has enormously increased across industrial and academic fields. In fact, it can provide better insights for the problem being addressed and it helps mitigate risk and make smart decision by proper risk analysis. Thus, the analysis of such data has tremendously grown making the scientists to tap the dark data that was considered useless few decades ago. Health is surely one of the areas that will benefit the most from the exploitation of the ever-increasing amount of

data that are recorded every day in hospital services, medical practices and among health professionals. The volume of this data as well as their inherent complexity make clinical decision-making more challenging than ever for clinicians and other care givers. Moreover, extracting knowledge from "Big Data" remains a challenging task. However, in addition to the problem of managing and securing these data, they have a real potential to facilitate the work of clinicians, in particular by setting up tools to assist in decision-making.

2.1 Medical Archives

With the advances in technology related to medical signals and image acquisition, there has been an escalation of complexity in medical data which has opened new opportunities for the researchers to reform the modern medicine. These advances have begotten medical innovations, such as navigation and monitoring systems, novel imaging technologies and revolutionary surgical tools (magnetic resonance imaging (MRI), ultrasound imaging, surgical microscope, etc.). These medical devices are becoming more versatile, and they are collecting and analyzing more data than ever before, resulting in massive digital medical databases relatively unexplored. [Healthcare, 2008] states that the number of images acquired in the United States in the radiology and cardiology departments increases at annual rate of 6 to 8 % during the last decade. This would represents a volume of about 100 petabytes in 2014, which is tantamount to the data on the servers of Facebook at that time. Digital archives are able to combine a very large number of clinical cases and be a rich source of information, but difficult to exploit directly by clinicians. It can be very time consuming and daunting to go through all archived cases. Therefore, the automatic interpretation of these archives is an essential step in the development of methods of medical decision-making. In this thesis, we are interested in the digital data coming from the operating room (OR). This data has spurred the community to build a context-aware system (CAS) which treats the information available in the OR to provide contextual support to the clinicians. The work done to date, such as [Cleary et al., 2005] [Bharathan et al., 2013], are ranging from simply showing the relevant information appropriately during surgery to assisting the clinicians in performing challenging surgical tasks by providing recommendations/warnings or suggesting actions to take. To provide such a support, it is required to use efficient tools to discover patterns in the large data sets which involves methods at the intersection of machine learning, statistics, and database systems. This is represented by data mining. In the following sections, we describe in short the data mining approach. Then, we introduce the medical applications issued from analyzing the medical data. Ultimately, we summarize this section along with introducing the field of interest of this thesis.

2.1.1 Data Mining

Data mining is the process of discovering meaningful knowledge, such as patterns, associations, changes or significant structures from the massive amount of data. It finds patterns in data that probably human would not find. As illustrated in

Fig. 2.1, it brings together all the methods allowing the transition from of the raw data to the knowledge of a domain. Different approaches are possible, the first is to automatically describe the raw data. This is the case in the clustering methods, that seek to automatically gather data into separate groups. A second type of approach is the association rule mining. It is a pattern that states when an event occurs, another event occurs with certain probability. An example of the latter approach is the time-series analysis (e.g. the Apriori-like technique [Huang et al., 2000]) which consists of finding particular regularities and features, including mining sequential patterns and periodicities, and search for similar sequences. Another type of approach is to use the data and the knowledge associated with them to predict or explain one or more observations. These approaches are based on supervised machine learning algorithms, such as decision trees or neural networks.

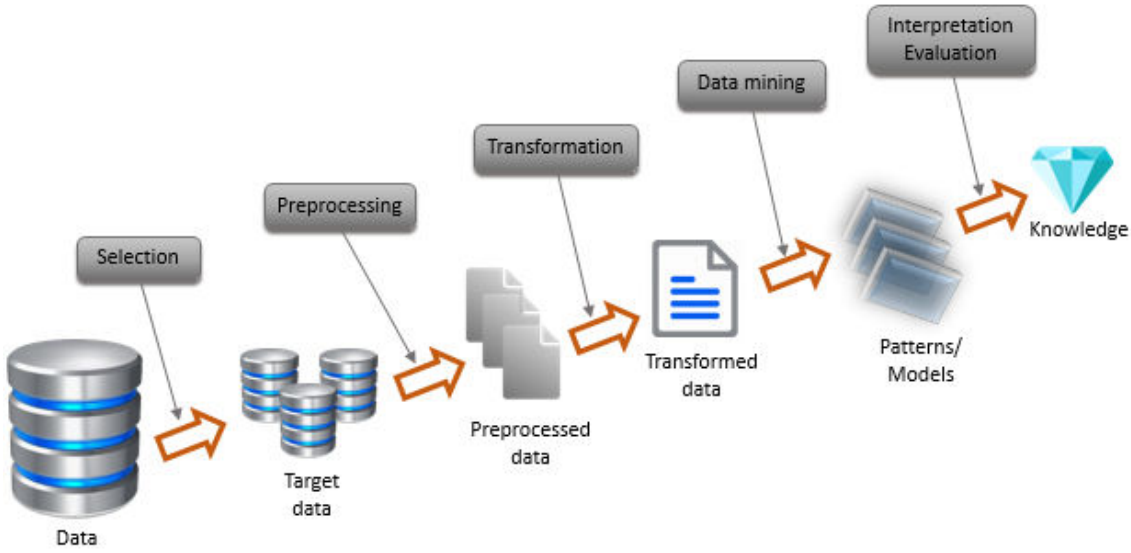


Figure 2.1: The transition from raw data to knowledge.

Data mining applications are very large [Padhy et al., 2012] [Liao et al., 2012] and are widely used in diverse areas. These applications are based on vital methods (predictive and descriptive models: classification, regression, clustering etc.) adapted to the large amount of stored data and the resources available to analyze and exploit them automatically. In the medical field [Hashemi et al., 2018], data mining is mainly used in the context of epidemiology, i.e. the study and analysis of causes, patterns and effects of health and diseases.

2.1.2 Computer-aided Decision in Medical Imaging

Nowadays, medical devices produce large volumes of data per patient in seconds, making it tedious for clinicians to quest the information while providing timely diagnoses. This presented a significant need for development and improvement of computer-aided decision support systems in medicine. These systems are used to integrate clinical and patient information and provide support for decision-making

in patient care. Also, they are designed to assist the clinicians and other health professional in choosing between certain relationships or variables for the purpose of making a decision [Chen et al., 2013]. This situation calls for the use of medical science methods to process the massive amount of data and construct a computer-aided decision systems to assist such decision makers. In fact, there are several areas in medicine for which computer-aided decision systems have become implemented and designed. In the following sections, we describe in short two different types of computer-aided decision and their applications.

2.1.2.1 Computer-aided Diagnosis

One way to reuse the medical archives is to look at the medical records that have commonalities between patients who have the same diseases. This improves the medical knowledge by identifying new diagnostic rules that will be taught to clinicians or directly to patients. Using the medical records containing complex digital data, such as images or videos, it is possible to extract diagnostic rules that link the patterns identified in the data to a diagnosis. However, it is not necessarily obvious to teach the clinicians new diagnostic rules based on numerical features, such as the texture of an image or a motion feature in a video. In this context, rather than trying to teach complex rules to clinicians, it is preferable to train automatic decision algorithms based on medical records. So, computer-aided diagnosis (CAD) is a form of the employment of machines to support human diagnostic reasoning. Several studies, such as [Doi, 2007], have suggested that the incorporation of the CAD system into the diagnostic process can improve the performance of image diagnosis by providing the quantitative support for the clinical decision. The purpose of CAD system is then to improve the diagnostic accuracy and the consistency of clinicians' interpretation by using the system output as a guide. It is necessary to note that the CAD system is used only as a tool to provide additional information to clinicians who will make the final decision as the diagnosis of the patient. Usually two types of general approaches are employed in computerized schemes for CAD systems. One is to find the location of lesions by searching for the abnormal patterns. Another is to quantify the image features of normal and/or abnormal patterns. CAD is applicable to all imaging modalities, including projection radiography, computer tomography (CT), MRI, ultrasound imaging, and nuclear medecine imaging. In addition, computerized schemes for CAD can be developed for all kinds of examinations on every part of the body. The most popular application is probably the automated breast cancer screening in mammograms [Shin et al., 2015], which allows to some extent to replace a second medical opinion by an automated diagnosis. The work of the team in this context was concentrated in the diabetic retinopathy (DR) which is detailed in the section 2.1.3.1.

2.1.2.2 Computer-assisted Surgery

In addition to the diagnostic assistance, digital medical records can also be reused for surgical assistance. Computer-assisted surgery represents a surgical concept and a set of methods, that use computer technology for surgical planning and for guiding or performing surgical interventions. Different types of computer-assisted

surgical systems can be used in the OR to support the clinicians. In our case, we are relying on the analysis of videos recorded during the surgery (via a microscope or an endoscope). In this context, the idea is to analyze the video stream in order to recognize the surgeon's actions, thus automatically analyze the progress of the surgery. The capability to automatically recognize the surgical task/activity plays a crucial role in the development of a CAS. Specifically, such a system would open up the possibility for many applications, both intraoperative and postoperative, as detailed in the following sections.

2.1.2.2.1 Intraoperative Applications

The intraoperative recognition of surgical tasks in the OR can be used to determine the information required by the clinician's team during the surgical procedure. For example, in the cataract surgery, the system could help the surgeon in identifying the orientation of the implant (the artificial lens that replaces the natural lens of the eye) while injecting it in the eye. This would help to get rid of employing two tools dedicated to this task, leading to diminution of the procedure time. Furthermore, if one is able to analyze the surgical video streams in real-time, it is then possible to train automatic decision systems: as soon as an abnormal event is detected during the surgery, an alert can be generated. Also, it can be seen as real-time notification system [Twinanda et al., 2017] that calls senior surgeons when certain key surgical activities are being executed in the OR or in the presence of a particularly critical situation. Moreover, it is possible to inform the surgeons of defects identified in certain surgical operations, for instance the system would suggest the best actions to handle such critical cases. So, surgeons can then adapt the workflow in order to improve the safety and the effectiveness of the surgery. It is then possible to develop semi-automatic decision support tools: real-time recommendations can be generated by relying on similar surgeries within an archive [Charrière et al., 2017]. The kind of alerts or recommendations used in such scenarios should be informative, convenient and the simplest possible so we don't add complexity to the OR. In addition, these real-time systems can optimize the surgical workflow and the OR resource management [Bhatia et al., 2007]. For instance, by knowing which surgical task is being done in the OR, the completion time of the surgery can be estimated. This can be used to notify the clinical staff to prepare the next patient [Doebbeling et al., 2012].

2.1.2.2.2 Postoperative Applications

The automatic recognition of surgical tasks is not only advantageous during the surgery procedure, but also afterwards when the surgery is over. For example, with the ability to automatically analyze the surgery, it can be possible to automatically generate a surgical report. In other words, events, actions and critical situations could be automatically identified, helping in the generation of the operative report. Using this report, it is also feasible to assess the surgical skill of the surgeons and track their improvement over time [Reiley and Hager, 2009] [Reiley et al., 2011]. In the training context, it can be used to accelerate the learning curve of the surgeons by

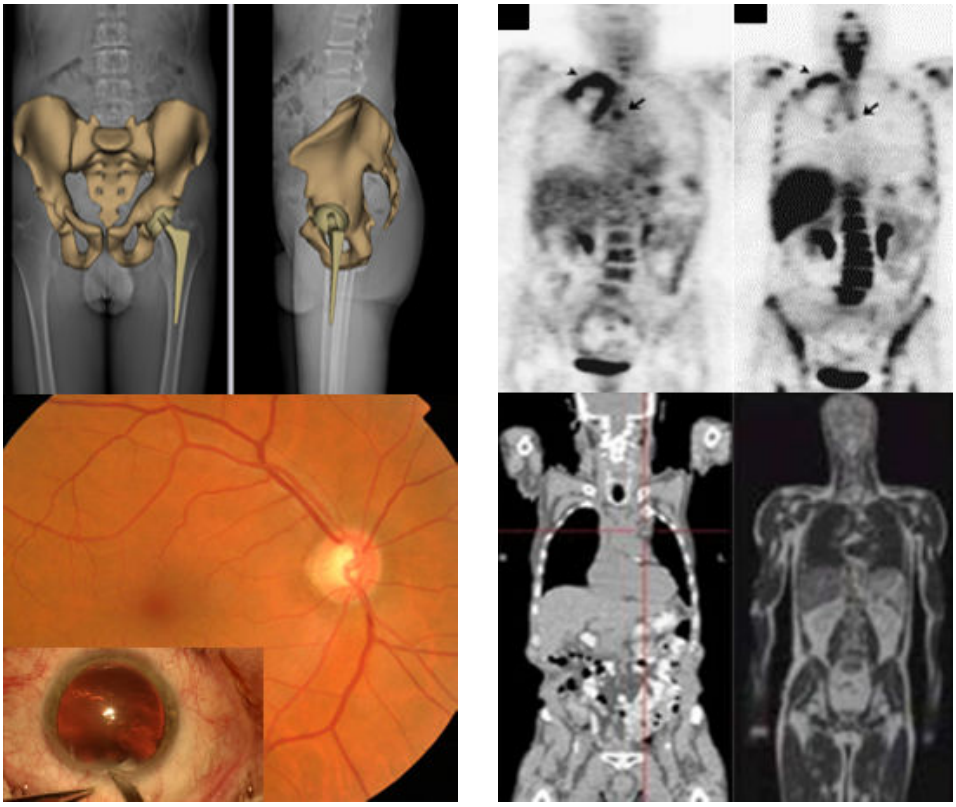
indexing the content of videos appropriately, thus faster access to surgical activities done by experts. Not to mention the possibility of improving the medical knowledge by retrospectively analyzing the workflow of the surgery.

2.1.3 LaTIM Research Positioning

The Laboratory of Medical Information Processing (LaTIM), UMR 1101 of INSERM (National Institute of Health and Medical Research), develops a multidisciplinary research (Fig. 2.2) in which information sciences and health sciences enrich each other through exchanges between the two fields. Inside the IMAGINE team (Multimodal Information Integration for Decision Support and Optimization of Interventional Therapy), the METIS axis (Multimedia mEdical informaTion analysIs, protectIon and Secondary use) develops research on medical databases for two aims: to secure the sharing of the medical data and to reuse them for medical decision support. In this context, several studies have been carried out in the field of content-based image retrieval (CBIR [Long et al., 2009]) and multiple-instance learning (MIL) [Quellec et al., 2017a]. Research on still images was then extended to the analysis of clinical cases containing image data and demographic data. In parallel, content-based video retrieval (CBVR [Hu et al., 2011]) and MIL studies were initiated in order to provide the per-operative support. The methods developed focused on the ophthalmology applications thanks to a strong collaboration with the ophthalmology department of the Regional University Hospital Center (CHRU) in Brest.

2.1.3.1 Works on Still Images

Regarding still images, the main objective of the work carried out at LaTIM is to assist in the diagnosis of diabetic retinopathy (DR). Diabetic retinopathy is a complication of diabetes reaching the retina. The diagnosis and the detection of this disease is made by a examination of the fundus of the eye. A large number of fundus images are then examined each year by ophthalmologists, in order to detect the presence and the number of possible lesions. In order to facilitate and accelerate the examination of these images, many automatic image analysis algorithms have been studied. The literature is very rich on this subject [Amin et al., 2016]. The early studies were based on wavelet transform [Quellec et al., 2008] [Quellec et al., 2010a] [Quellec et al., 2012b] [Quellec et al., 2010b]. Quellec et al. have also worked on decision-making methods by merging information from the images, with contextual semantic information such as age, sex or patient history [Quellec, 2008]. This method yields good performance with an acceptable error rate. Afterwards, methods based on multiple-instance learning were proposed in [Quellec et al., 2012a] [Quellec et al., 2011] [Quellec et al., 2016b]. They showed significant improvements compared to the previous studies done in the team. The multiple-instance learning concept was also used to automate the mammography examination [Quellec et al., 2016a]. In the last few years, the deep learning has emerged at breakneck speed. This led to a numerous studies trying to automatically diagnose the DR. For instance, at Google, [V et al., 2016] they achieved a new state of the art performance in the automatic diagnosis of DR using a deep learning based system. Also, a machine learning challenge [Kaggle, 2015] was organized with the aim to design an automated system for grading the



(a) Multimodal information integration for decision support and optimization of interventional therapy

(b) Therapeutic action guided by multimodal imaging in oncology

Figure 2.2: LaTim teams

severity of diabetic retinopathy (DR). The leading solutions were all based on deep learning. [Quellec et al., 2017b] has recently proposed a framework based on deep learning to detect automatically the DR and the lesions related to it. This framework has shown significant performance improvement compared to the previous work of the team.

2.1.3.2 Surgery Videos Analysis

The team has been interested for several years in the exploitation of videos recorded during surgeries such as cataract surgery in ophthalmology. The methods developed have a long-term objective of providing real-time assistance to the surgeon, i.e. offering examples of similar situations, recommendations or alerts. It is therefore necessary to be able to analyze in real-time the videos recorded during the surgery and to compare it to the data archived. To reduce and facilitate the search for similar cases, we rely on surgical workflows, which define the different steps of the surgery. An essential step toward this aim is to be able to recognize the surgical task being performed by the surgeon. As for still images, the methods studied are content-based methods, but this time on video sequences (using CBVR and

multiple-instance learning methods). Initially, various methods were developed to automatically recognize the surgical task performed within a sub-sequence. Then, these methods were adapted to perform a more complex task: automatic sequencing of a complete surgical video in surgical tasks. These methods are detailed in section 2.2.2.

2.1.4 Summary

Apart from medical data being inherently complex, the sheer volume of the medical data collected is growing speedily. Taking advantage of the technical and computerized trends, several medical applications have been introduced in the context of computer-aided decision. In particular, one of the applications is to analyze the archived surgical data to provide information to the surgeons in real-time. To this end, several effective methods have been developed around the reuse of medical data for surgical decision support. For the objective of real-time surgical support, several tracks have been explored and validated to allow the automatic surgical video analysis. This includes robust methods to recognize the surgical task/activity during the execution of the surgery. This is an essential step in order to be able to generate appropriate recommendations/warnings and recognize critical situations. In the coming section, we discuss the existing approaches that tackle the surgical activity recognition (SAR) domain, where we refer to the step/phase/task as an activity. Ultimately, we brief the position of our work with respect to the work already existing in the literature.

2.2 Activity Recognition

The need of SAR has emerged along with the interest of developing computer-assisted surgery systems. Various kind of signals got from the surgical equipments was used to tackle the problem of recognizing the surgical activity [Pernek and Ferscha, 2017]. One of the first applications based on activity recognition was to know the state of the OR, e.g. occupied or not occupied in [Bhatia et al., 2007]. In recent years, [Twinanda et al., 2017] [Charrière et al., 2017] [Dergachyova et al., 2016] have been shown that the surgical tool usage signals can provide valuable information in performing the activity recognition task in cholecystectomy and cataract surgeries.

In this thesis, we are interested in the surgical video analysis field. We focus the discussion in the following sections on vision-based approaches used to address the automatic surgical video analysis issue. First, we review the state-of-the-art methods for activity recognition in the computer vision domain. Then, we discuss the vision-based approaches that have been proposed in the medical domain. Afterwards, we describe the methods that have been proposed in the surgical tools recognition field.

2.2.1 Computer Vision Domain

Various video modalities have been used to tackle the activity recognition task. It was most commonly performed using radio-frequency identification (RFID), kinematic sensors, and video recordings of the operative area. These data are present in

large quantities and are difficult to exploit manually. It is then necessary to analyze them in real-time to recognize and anticipate abnormal situations. This is difficult to achieve and requires a large number of people to analyze different video sources in real-time. Many methods have been developed for the automatic analysis of this type of videos, especially for the detection of abnormal events and situations. Due to its importance for several domains, the activity recognition is used for a wide variety of applications, such as ambient assisted living for smart homes [Bilinski et al., 2013], health care monitoring solutions [Doulamis et al., 2010], security and surveillance applications [Xu et al., 2016], and tele-immersion applications [Roy et al., 2016]. The methods in this field are typically built by using a two-step pipeline: visual feature representation step and the activity recognition step.

The visual feature representation can be divided into two main groups: spatial and spatio-temporal features. The spatial features can be color information [Jain and Vailaya, 1996] and texture features [Manjunath and Ma, 1996]. They can be deemed as global descriptors. Also, there exist local descriptors which represents the characteristics of image patches, for instance key-points detectors and descriptors like Harris corner detector [Harris and Stephens, 1988], scale-invariant feature transform (SIFT) [Lowe, 2004], speeded-up robust features (SURF) [Bay et al., 2008], and histogram of gradients (HOG) [Dalal and Triggs, 2005]. They are to some extent invariant to background clutter, appearance, occlusions, and to scale and rotation in some cases. Then, these local descriptors can be combined to build a global descriptor using feature encoding methods, such as the bag-of-word (BOW) approach. And to provide semantic information, high-level features, such as human pose estimation [Xu et al., 2012] and the results of object detection [Wu et al., 2007] are then utilized for activity recognition. By using such high-level features, the methods will have better performance in modeling the activities performed in the scene. On the contrary, the spatio-temporal features consider not only the spatial information but also the temporal information from an image sequence, such as optical flow-based features [Chaudhry et al., 2009] and spatio-temporal features such as the spatio-temporal key-points detector proposed in [Laptev, 2005]. These features are referred to as handcrafted features, where the domain knowledge is used to extract features that makes the methods work. But, in recent years, features learning methods such as principal component analysis (PCA [Abdi and Williams, 2010]), independant component analysis (ICA [Parsons, 2005]) and artifical neural networks (ANN [Specht, 1988]) in particular deep learning, have earned a dearly interest in the computer vision field. Deep learning achieved a new state-of-the-art performance in different types of tasks in this field, for instance the activity recognition in [Tran et al., 2015]. One of the most common deep learning algorithms is the convolutional neural networks (CNN). In 2012, a CNN architecture in ImageNet challenge, referred to as AlexNet [Krizhevsky et al., 2012], reached a new state-of-the-art performance in classifying one thousand different classes. This network has shown its ability to learn discriminative learning features in order to do the task.

The second step of the activity recognition pipeline is the model/algorithm used to classify/detect/recognize the activity. Numerous methods were applied to the automatic detection of road traffic or pedestrian flows. [Piciarelli and Foresti, 2006] realized a structured trajectories partitioned in trees to achieve the automatic de-

tection of abnormal vehicle trajectories. Indeed, by detecting the typical movements and trajectories, it is easier to locate the abnormal events. Thus, [Hospedales et al., 2012] relied on probabilistic Bayesian methods to automatically analyze the behaviour of pedestrians or vehicles in real-time. This method allows to learn classical patterns of behaviour, thus leading to detect atypical events. These methods are efficient in distinguishing atypical events from the normal ones previously learned. However, surveillance videos are relatively different from surgical videos: they are generally filmed with a fixed background. In fact, the SAR methods can generally be categorized in two groups of methods: pre-segmented and frame-wise classification methods. The former is the task of labeling videos with their corresponding activity labels. The latter consists of identifying a sequence of activities performed in a video without any information regarding the beginning/ending of each activity. On the one hand, classification algorithms can be used to address the pre-segmented classification such as discriminative modeling (SVM in [Xia and Aggarwal, 2013]). One the other hand, the frame-wise classification requires the incorporation of the temporal information in the recognition pipeline. Dynamic time warping (DTW), hidden markov model (HMM) and dynamic Bayesian networks (DBN) are the most well-known approaches used to address this problem.
Among deep learning methods, various approaches have been recently proposed which were inspired by the two-stream CNNs proposed by [Simonyan and Zisserman, 2014a]. It incorporates spatial and temporal information extracted from RGB and optical flow images. These two image types are fed into two separate networks, and finally they fused the prediction score of each network. This method is the basis of many other methods such as [Tran et al., 2015] [Sun et al., 2015] [Zhu et al., 2016] [Yue-Hei Ng et al., 2015] [Feichtenhofer et al., 2016]. In fact, CNN can extract spatio-temporal features but only on a fixed-length of image sequences. To incorporate the temporal information inside the CNN, recurrent neural network (RNN) architecture was proposed in [Werbos, 1990] [Rumelhart and McClelland, 1987]. However, it was demonstrated in [Bengio et al., 1994] that the RNN are difficult to train when the gap between the relevant information and the point where it is needed is very large. This is related to the vanishing weights problem [Bengio et al., 1994]. To overcome this issue, a modified architecture called Long Short-Term Memory (LSTM) was proposed in [Hochreiter and Schmidhuber, 1997]. [Donahue et al., 2017] has proposed a combination of a CNN and LSTM, which performs fairly well on various tasks, e.g. image description and activity recognition. However, traditional two-stream CNNs are unable to exploit the correlation between the spatial and temporal streams. In [Ma et al., 2017], spatial and temporal features were extracted from a two-stream ConvNet using *ResNet-101* pre-trained on ImageNet, and fine-tuned for single-frame activity prediction. The spatial and temporal features are concatenated and then used as input to: Temporal Segment LSTM (TS-LSTM) or Temporal-Inception. This method yields a new state-of-the-art performance in the activity recognition domain. A detailed technical explanation about CNN and LSTM is presented respectively in sections 5.1 and 6.1.

2.2.2 Medical Domain

In the last decade, the analysis of videos has started to appear in the medical field, in particular for the analysis of endoscopic and laparoscopic videos [Padoy et al., 2012] [Twinanda et al., 2017], analysis of surgical scenes [Piciarelli and Foresti, 2006] [Blum et al., 2008] [Donahue et al., 2017], and lately analysis of cataract surgery videos [Quellec et al., 2014a] [Quellec et al., 2015] [Charrière et al., 2017]. The methods of this field have goals ranging from indexing medical video datasets to provide a real-time assistance to the surgeons during the surgery. Yet, there are still few methods proposed in the literature on the automatic analysis of surgical videos and the SAR is no exception to this matter. The scarcity is due to several reasons. First, the video acquisition is challenging because of the unavailability of the required equipments to do it and the strict regulations applied inside the OR. Also, this topic is relatively a new research field comparing to the activity recognition in the computer vision field. In addition, there are numerous visual challenges in the surgical videos, e.g. occlusions, rapid camera motions and reflections. In fact, there are only few surgical activity datasets that have been released publicly, e.g. *JIGSAWS* [1], *EndoVis* [2] and *m2cai16-workflow*[3] datasets.
In this context, several vision-based studies have been proposed to tackle topics relatively related to the activity recognition task, such as surgeon skills evaluation [Suzuki et al., 2015], surgical tool recognition [Twinanda et al., 2017]. Precisely, in this thesis, we address the surgical tool recognition problem which is a fundamental element in the SAR task, thus in any computer-assisted surgical system.

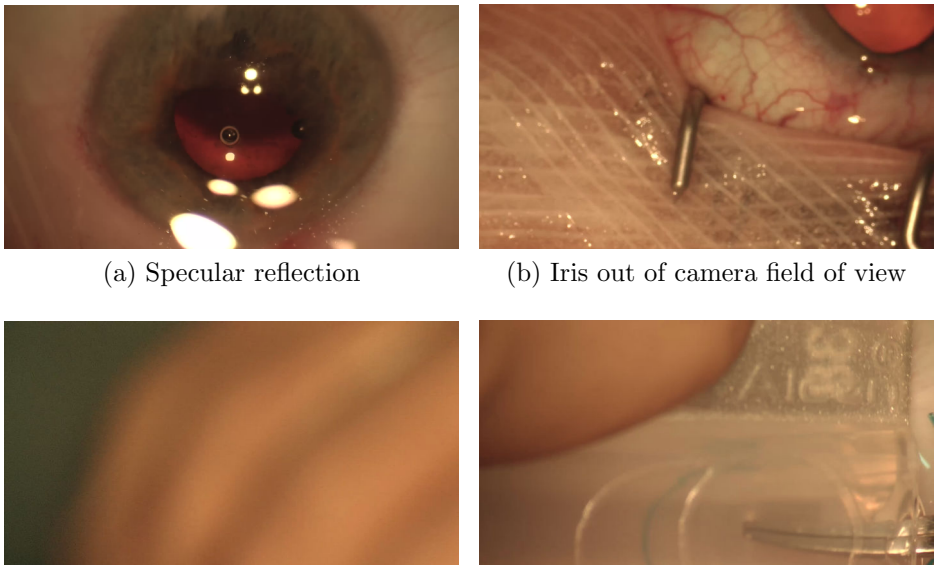
2.2.2.1 Visual-based Representation

Different types of signals exist in the OR. In this thesis, we are interested in the visual signal emanating from the surgical equipment filming the operative scene. One of the biggest advantage of the surgery videos analysis is that it does not require any installation of additional components in the OR that would alter the surgical procedure. But, this analysis would require to overcome the visual challenges presented in these videos, such as the challenges shown in Fig 2.3. For further details, the dataset generated in this thesis and the challenges presented in it are detailed in section 3.3.

2.2.2.2 Surgical Workflow

In order to design useful computer-aided systems, we need to define what the algorithms should look for in videos. In particular, if we want the algorithms to automatically extract the workflow, we need to establish a terminology for describing this workflow and provide visual examples for each term. In the literature, there are different ways of describing a surgery. A surgery can be defined at different levels of abstraction. This is called granularity levels. Depending on the granularity,

[1] https://cirl.lcsr.jhu.edu/research/hmm/datasets/jigsaws_release/
[2] https://grand-challenge.org/site/endovissub-workflow/data/
[3] http://camma.u-strasbg.fr/m2cai2016/index.php/program-challenge/



(a) Specular reflection (b) Iris out of camera field of view

(c) Blurred field of view (d) Occlusion challenge

Figure 2.3: Visual challenges in cataract surgeries.

various terms have been used to refer to the activity, such as surgeme [Lin et al., 2006], phase [Blum et al., 2010] [Padoy et al., 2012], and gesture [Zappella et al., 2013].

In a recent review of surgical activities definition [Lalys and Jannin, 2014], they presented different levels of granularity that can be found in the literature. The finest description level corresponds to the visual features extracted from the video, such as the presence/absence of object/tool. They are induced by actions or gestures made with surgical instruments. An action can be seen as the application of a gesture to realize an objective. The terms "steps", "task", or "phases" are also often found in the literature. A task corresponds to a job that must be carried out with a precise objective. It is therefore related to the realization of a surgical objective, such as making an incision or making a suture. The same task can be performed several times in the same surgery. We can define a "step" as a sequence of physical actions/tasks, which does not necessarily lead to the realization of a surgical objective. A set of steps make up a surgical phase. The term surgical phases corresponds in the literature to high-level surgical tasks. They must lead to the realization of an essential surgical objective for the surgery. At the end, the coarsest level of description is the surgical procedure itself. This level is used in the case where one seeks to differentiate automatically the type of surgery, or the examination carried out. For instance, [Twinanda et al., 2015] sought to automatically determine the type of abdominal surgery performed. Also, a multi-level granularity approach is possible and it has been explored in [Charrière et al., 2017] [Forestier et al., 2015].

2.2.2.3 Activity Recognition in Surgery Videos

In this thesis, we are dealing with two types of videos: tool-tissue interaction videos and surgical tray videos, described in details in the next section (Section 2.2.2.3.1). In contrast to the surgical tray, a wide variety of studies have addressed the SAR on the tool-tissue interaction videos. In this work, we are especially interested in the tool-tissue interaction videos along with the information coming from the surgical tray videos. In this section, we describe the methods proposed for the SAR on tool-tissue interaction videos.

One of the earliest studies in this field is [Lo et al., 2003]. A pipeline was proposed, based on a Bayesian network on top of visual cues related to shape and deformation changes and other low level features, to segment the laparoscopic videos. This method produced promising results in segmenting these videos. The following studies tended to use the tool usage signals with the assumption that this information can be obtained whether by further analysis of the video or through other sensors. [Ahmadi et al., 2006] [Padoy et al., 2007] [Blum et al., 2008] were based on DTW or HMM and produced satisfactory results for SAR. Afterwards, [Padoy et al., 2008] proposed a pipeline based on HMM on top of a combination of visual features and tool usage signals. This mixture yields promising results in recognizing the phases in laparoscopic videos. In [Blum et al., 2010], the tool usage signals were used only at learning time since it was difficult to obtain them at test time. The canonical correlation analysis (CCA) were used to reduce the dimension of the handcrafted visual features, expectedly leading to more semantically discriminative visual features. It was tested on laparoscopic videos. This method showed better performance than the PCA-based methods.

In addition, the motion information has been used as part of the visual features in the SAR studies. In [Zappella et al., 2013], in addition to HOG, the motion data represented by histogram of optical flow (HOF) around the detected STIPs were extracted to represent the video frames. But, these videos were recorded for training purposes so not as long as real procedures. Another work made by [Lalys et al., 2010] to extract the surgical phases in the microscope videos. This method is based on a multitude of visual features such as color, texture and shape information. Using the combination of SVM and HMM on top of the visual features, the pipeline yields promising result for the recognition of the phases. In another work from the same team [Lalys et al., 2012], similar features were used in a vision-based methods to segment the phases in cataract surgery. DTW was on top of the mixture of the features which has been proven to perform very well. In [Forestier et al., 2015], they developed a real-time method for automatic video annotation. They wanted to recognize which surgical phase is being performed by the surgeon, relying on a low granularity level (surgical actions). This method used decision trees to model the surgical workflow instead of HMM. The visual content of the surgery videos was not analyzed: the features used as input of the model was the ground truth of the action being carried out by the surgeon. Recently, in [Dergachyova et al., 2016], HMM based pipeline was proposed to address the surgical phase recognition on laparoscopic videos. The visual features extracted are color, texture and shape information. The results prove similar trend as in [Charrière et al., 2017] that tool usage signal are better than the visual features in performing the task.

But, the combination of tool usage signals and visual features produced the best performance in this method. It is also in accordance with [Padoy et al., 2008] where the combination of visual features and tool usage signals yields the best recognition results.

The aforementioned methods were all based on handcrafted features but a learning feature approach could be better representative, expectedly leading to better performance. In an early study [Klank et al., 2008], an approach based on genetic algorithm to learn feature representations for surgical phase recognition on laparoscopic videos was proposed. A SVM model was on top of the extracted features to obtain the recognition results. The results showed that the learnt features are more discriminitave than the handcrafted ones. Recently, the deep learning based methods have rapidly emerged. But, due to the limitation of the number of large public dataset in the medical field, few deep learning approaches were proposed to tackle the SAR on tool-tissue interaction videos. In [Lea et al., 2016], CNN based approach is proposed to perform action segmentation. This method has been evaluated on JIGSAWS dataset. It was based on spatial and temporal convolutional components. The results proved that this approach is better than other deep architectures and handcrafted-based features pipelines. Recently, solutions based on RNNs have also been proposed [Jin et al., 2016] [Bodenstedt et al., 2017] [Twinanda et al., 2016]. These RNNs process instant visual features extracted by a CNN from images. In particular, [Jin et al., 2016] apply a CNN+LSTM network to a small sliding window of three images. [Bodenstedt et al., 2017] apply a CNN+GRU network to larger sliding windows and copy the internal state of the network between consecutive window locations. As for [Twinanda et al., 2016], they apply a CNN+LSTM network to full videos. Interestingly, the CNN proposed by [Twinanda et al., 2016], namely EndoNet, detects tools as an intermediate step. A challenge on surgical workflow analysis was organized at M2CAI 2016[4]: two of the top three solutions relied on RNNs, more specifically on LSTM networks [Jin et al., 2016] [Twinanda et al., 2016]. In [Twinanda et al., 2017], an end-to-end approach was proposed to automatically detect the presence/absence of the surgical tools and the surgical activity at once on laparoscopic videos. The CNN was based on an extended version of AlexNet architecture. The features extracted by the CNN were fed to a SVM model to do surgical phases recognition. Also, a HMM model was on top of it to add the temporal constraints. This method performed very well in recognizing the surgical phases in laparoscopic videos. They also demonstrated that a transfer learning approach, pre-trained on ImageNet dataset for example, significantly improve the results.

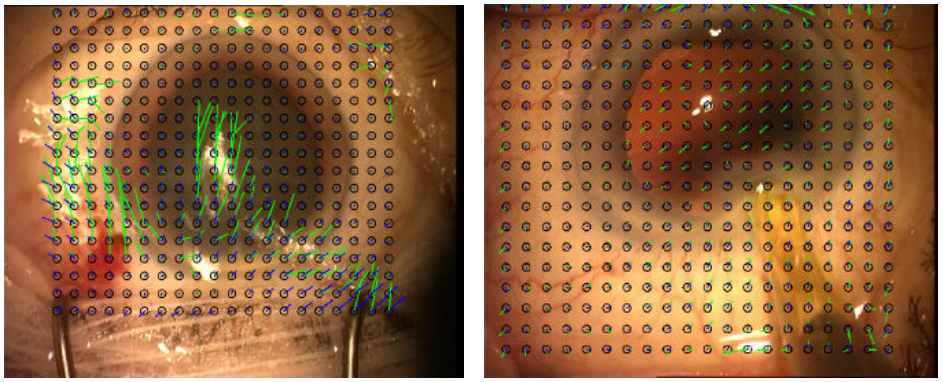
As demonstrated in the latest pipelines proposed in this field, the tool usage signals contain strong discriminative and semantic information. Both handcrafted and features learning approaches proved that the tool usage signals is a key ingredient to any automatic SAR, thus the need to detect/recognize the surgical tools.

2.2.2.3.1 LaTIM Work on Surgery Videos Analysis

At LaTIM, The work in the field of surgical video analysis currently applies to the cataract surgery. In a first step, several methods have been developed to auto-

[4] http://camma.ustrasbg.fr/m2cai2016/index.php/workflow-challenge-results/

matically recognize a surgical task performed within a subsequence. These methods are based on CBVR and MIL approaches. Then, these methods were adapted to automatically annotate videos of complete surgeries. The videos are then sequenced automatically in surgical activities. In this section, we describe the various methods developed at LaTIM towards the target of automatic SAR.
One of the first methods was [Droueche, 2012] which was based on visual features derived from MPEG compression (Moving Picture Experts Group). Extented Fast Dynamic Time Warping (EFDTW) algorithm was used to measure the similarity between two complete videos. The results were encouraging but this method was computationally expensive and therefore does not allow real-time assistance. The experiments were done on two different ocular datasets: membrane peeling and cataract dataset.
To simplify the problem, rather than analyzing a thorough surgical video, the problem has been reduced to the classification of video sequences. The videos were cut into sequences where each of which represents a surgical activity. In this context, several methods adapted to real-time have been developed. [Quellec et al., 2014a] proposed to automatically cut out a surgical activity in elementary motions to facilitate its recognition. The subsequences were described by vectors invariant to the variations in duration and speed of execution within surgical tasks. Then, they were compared to the archived subsequences. This system provided a very fast solution and good recognition rates.

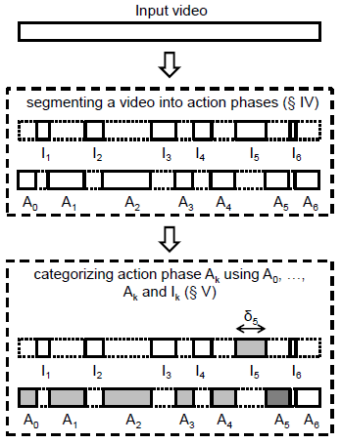


(a) courtesy to [Quellec et al., 2015] (b) courtesy to [Quellec et al., 2015]

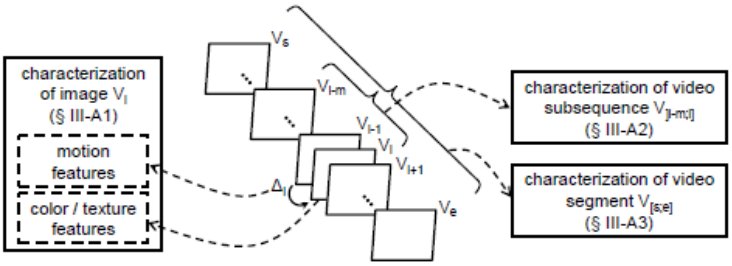
Figure 2.4: In blue, the motion fields approximated by spatio-temporal polynomials. In green, the motion fields between two consecutive images measured by the Farnebäck algorithm [Quellec et al., 2015].

A second method [Quellec et al., 2015] consists to approximate the displacement fields by a spatio-temporal polynomial during a short video sequence, as illustrated in Fig.2.4. For each surgical activity, a MIL process is performed to identify which spatio-temporal polynomials are extracted when this activity is performed in the video sequence. The same concept was applied for query video sequence in order to identify the surgical activity performed. The experiments were done on a cataract video dataset and the results were very well in recognizing the surgical activity. However, these two approaches were not realistic, as they required manual segmen-

tation of the videos (to indicate the beginning and the end of each surgical step). The next step was therefore to set up methods for segmenting these steps automatically. A real-time method of automatic segmentation in surgical activities has been proposed by [Quellec et al., 2014b]. This method is based on the fact that there is usually a transition period between two surgical activities (called idle activity), in which there is nothing taking place in the microscope field of view. This delay comes from the fact that the surgeon changes tools between two surgical tasks. The



(a) courtesy to [Quellec et al., 2014b]



(b) courtesy to [Quellec et al., 2014b]

Figure 2.5: In (a), the method of segmenting and categorizing of subsequences proposed by [Quellec et al., 2014b]. In (b), the activity recognition approach proposed by [Quellec et al., 2014b].

proposed method works initially on the detection of these transitions, based on a nearest neighbours cases approach. The surgery is then segmented temporally into "action phase" and "transition phase". Whenever a "transition phase" is detected, the "action phase" preceding it is classified (As shown in Fig.2.5a). Conditional Random Fields (CRFs) were used for the classification part. The motion features between the current image and the previous one, color and texture features as well as duration of the "transition phase" are used to build the visual signatures of the segments (As shown in Fig.2.5b). This method were evaluated on a cataract video dataset and the performance was fairly good. But, one main limitation in this approach is that several surgical activities can take place during the same action phase. This is the case if the "transition phase" is not detected between two surgical activities or if the surgeon has changed the tool from one hand to another while continuing the same action.

Lately, [Charrière et al., 2017] have proposed improvements to the previous work done in the team. At the outset, they relied on content-based search approach to automatically recognize the surgical activity in a subsequence of a surgical video. This part was based on the motion features extracted from the subsequences. Then, the method proposed in [Piciarelli and Foresti, 2006] was adapted to compare the subsequences. This methods yields fairly good performance in the SAR in the subsequences. Afterwards, a real-time method was proposed to segment a thorough sur-

gical video into activities. The methodology of this method is illustrated in Fig 2.6. The method models the surgical workflow by using both the relationships between the different granularity levels proposed in this work and the temporal relationships that exist between the labels of each granularity level. This model has been evaluated with two granularity levels: the steps and the phases. For the modeling of the

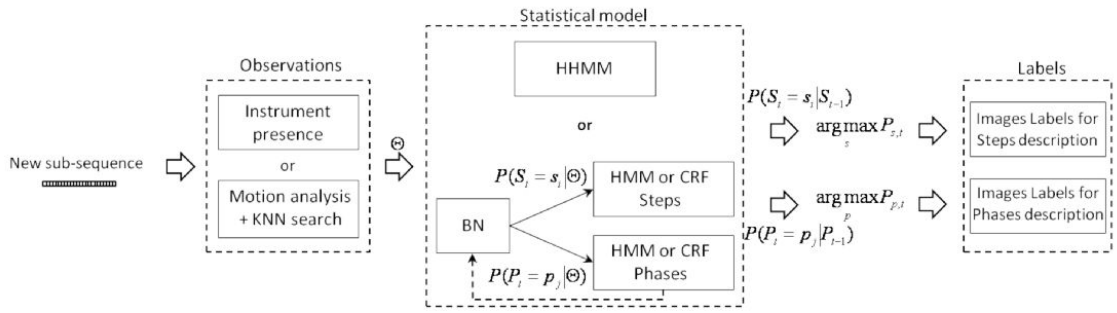


Figure 2.6: courtesy of [Charrière et al., 2017].

relation between the granularity levels, a Bayesian network was used. For modeling the temporal workflow of the surgery, a Hidden Markov Model (HMM) or a Conditional Markovian Field (CRF) was used for each of granularity level. Two types of observations were evaluated: tool usage information in the microscope field of view (manually annotated) and the motion information. Satisfactory performance was obtained by using the motion analysis information. Very good results were obtained when using the tool usage information as an input for the system. These results were particularly obtained with CRF and they motivate the automatic recognition of surgical tools.
In fact, the automatic recognition/detection of surgical tools can be accomplished in two ways: vision-based surgical tools detection/recognition methods or using methods not based on real-time surgical video analysis, such as RFID. For asepsis reasons, it is difficult to add external elements to the surgical tools. Then, it is advantageous to do the analysis using vision-based methods because of the strict regulations applied in the OR.

2.2.2.4 Surgical Tool Detection

The latest progress done in the SAR field has spurred the community to address the surgical tool recognition problem due to the strong correlation between surgical activities and tool usage signals. This topic is not only closely related to surgical activity recognition but also can be used in various applications, such as surgical video indexing and surgical report generation. Therefore, several tool recognition techniques have been proposed in recent years [Bouget et al., 2017]. In fact, they have addressed tool detection [Reiter et al., 2012], localisation [Allan et al., 2013] and pose estimation [Allan et al., 2014] using different types of cues and classification strategies on tool-tissue interaction videos. For instance, employing geometry information about the tools [Pezzementi et al., 2009], markers [Reiter et al., 2011], 3D coordinates of the insertion point [Voros et al., 2007], fusing kinematic and visual

information [Reiter et al., 2014] and through multi-class pixel-wise classification of tools position, color and texture features with different kind of machine learning techniques such as Boosted Decision Forests [Bouget et al., 2015] and Random Forests (RF) [Allan et al., 2013]. On the surgical tray, a limited number of studies have tried to detect the surgical tools. Most of them are not vision-based methods such as [Meißner and Neumuth, 2012], in which the RFID signals coming from the tools are used with a HMM in order to predict the tools being used by the surgeons. However, in [Glaser et al., 2015], a complicated system of two cameras and a scale is used to combine the weights of the tools and the visual features extracted from the images in order to detect the tools presence over the tray.

In [Rieke et al., 2016], a real-time method to track the surgical tools was proposed. They addressed the problem in two stages: tracking and pose estimation. They employed random forest for both stages. The tracking random forest employs RGB intensity information whereas the pose estimation forest uses HOG features for estimating the locations of the tool. In the first step, the tracker reduces the region of interest to a rectangular area around the tool tip by relating the motion of the tool to the induced changes on the image intensities. In the second step, a gradient-based pose estimation algorithm estimates the location of the instrument parts inside the bounding box. This method yields very good performance in tracking the surgical tools on three different datasets. However, this method is designed for only one single tool at a time, whereas, in our case, we have a plenty of tools on the tray and up to three tools in the tool-tissue interaction videos.

A tool detection challenge was organized at the M2CAI 2016 workshop: the objective is to identify all surgical tools that are present in each image of the laparoscopic videos. The dataset used in this workshop is the *m2cai16-tool* dataset. It consists of 15 cholecystectomy videos with ground truth binary annotations of the present tools. They have defined seven surgical tools that are typically used in cholecystectomy procedures. Following the trend in medical image and video analysis [Shen et al., 2017], the best solutions all relied on CNNs [Raju et al., 2016] [Sahu et al., 2016] [Twinanda et al., 2017] [Zia et al., 2016]. These best solutions relied on a transfer learning strategy: well-known CNNs trained to classify still images in the ImageNet dataset were fine-tuned on images extracted from surgery videos. In particular, [Sahu et al., 2016] and [Twinanda et al., 2017] fine-tuned AlexNet [Krizhevsky et al., 2012], [Raju et al., 2016] fine-tuned GoogLeNet [Szegedy et al., 2015a] and VGG-16 [Simonyan and Zisserman, 2014b], and [Zia et al., 2016] fine-tuned AlexNet, VGG-16 and Inception-v3 [Szegedy et al., 2016b]. However, the temporal information was not exploited in these solutions, like in all other state-of-the-art tool detection algorithms [Bouget et al., 2017].

In [García-Peraza-Herrera et al., 2016], they tried to exploit deep learning along with the optical flow in order to produce accurate segmentations of highly deformable tools. This method was based on Fully Convolutional Networks (FCN) proposed by [Long et al., 2015]. This network was adapted to do the segmentation instead of classification in two steps: the fully connected layers were replaced by convolutional layers to preserve the spatial information and the deconvolution layers were employed to generate the output of the segmentation. Since the FCN is not compatible with real-time processing, they use the optical flow to register the last segmented frame

with the current one when the FCN is busy processing a frame. This method yields fairly good performance in segmenting the surgical tools on the *EndoVisSub* dataset. In a more recent work, [Garcia-Peraza-Herrera et al., 2017] have proposed a framework called ToolNet to segment the robotic surgical tools. It consists of two lightweight architectures: ToolNetMS and ToolNetH. ToolNetMS aggregates multi-scale predictions and uses the Dice score as a loss function. Inspired by the holistically-nested edge detection [Xie and Tu, 2015], ToolNetH uses Multi-Scale Dice Loss (MSDL) that imposes multi-scale consistency and takes into account the accuracy of predictions at different scales. These methods show competitive results compared to other state-of-the-art methods.

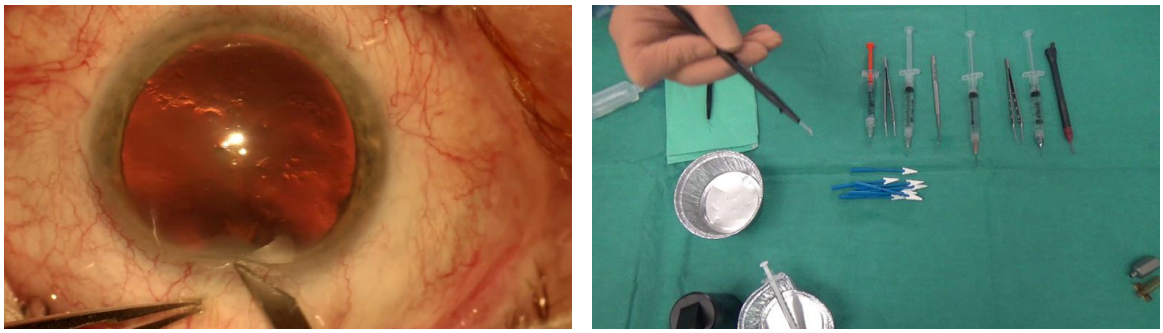
[Sarikaya et al., 2017] proposed a solution to the tool detection and localization problem in robot-assisted surgery (RAS) video understanding. The solution was based on Region Proposal Networks (RPN) jointly with multimodal two stream CNNs for object detection. First, two separate CNNs processed two different modalities: the RGB video frame and the RGB representation of the optical flow of the same frame as a temporal cues. They convolve the two input modalities to get their feature maps. Using the RGB input feature maps, they train a RPN to detect the possible tool regions. Thereafter, they continue the processing until the last fully connected layers where the features of both stream are fused before the classification layer. The experimentations showed competitive results to similar approaches.

Several recent works have been proposed to detect the surgical tools on the *m2cai16-tool* dataset. In [Wang et al., 2017], they trained VGG-16 and GoogLeNet separately, then they use ensemble learning to combine the results of the models to get the final results. This method outperforms the best solutions at M2CAI 2016 workshop. Also, [Choi et al., 2017] has recently proposed a pipeline based on a real-time object detection system "you only look once" (YOLO). It was first trained on the ImageNet dataset then fine-tuned on the images extracted from the *m2cai16-tool* dataset. This method yields very good performance in detecting the presence of surgical tools in the *m2cai16-tool* dataset. Inspired by the human visual attention mechanism, [Hu et al., 2017] have recently proposed AGNet to detect the surgical tool presence, which in turn outperforms the previous studies applied on the *m2cai16-tool* dataset. In AGNet, they have two sub-networks for surgical tool presence detection: global prediction network (GPN) and local prediction network (LPN). GPN was based on ResNet-101 architecture [He et al., 2016a]. In GPN, the network was modified to add a new convolution layer following the last convolution layer and they assigned the channel number for this new convolution layer as the number of surgical tool categories so they can obtain a visual attention map for each tool. They also computed a global prediction score for each tool by doing an average pooling on each map. Then, Otsu algorithm [Otsu, 1979] was employed to dynamically separate the pixels into categories: surgical tool and background. The minimum bounding box of the regions belonging to the surgical tool is used as input to LPN. The same network architecture was used for LPN except with using fully connected layer to produce the surgical tool predictions instead of just averaging the attention maps. A transfer learning approach was followed to initialize the weights of GPN, which was fine-tuned on the laparoscopic videos. Thereafter, they used the well-trained weights of the GPN as the pre-trained weights for the

LPN. The last step is to apply a gate function to obtain the final prediction results. Using AGNet, they achieved the best performance between the previous studies on detecting the tool presence in *m2cai16-tool* dataset, albeit they did not use the temporal information. However, [Mishra et al., 2017] was the first solution that incorporates the spatial and temporal information to detect the presence of tools in the *m2cai16-tool* dataset. ResNet-50 [He et al., 2016a] was used to extract the visual features from the videos, then a LSTM network was used on top of these spatial features extracted to capture the temporal connectionism across the visual features and thereby increase the accuracy in prediction. It was observed that the learning of temporal connectionism decreased the local error in detecting the tool presence. Therefore, this method is actually the state-of-the-art performance in terms of detecting the tool presence in *m2cai16-tool* dataset.

2.3 Thesis positioning

In this thesis, we address the problem of surgical tools recognition which is a complementary information to any SAR system. We propose to only use the visual signals emanating from the various equipments essential for the execution of the surgery. Although the tool usage signals is proved to perform better than the visual features in recent works [Charrière et al., 2017] [Dergachyova et al., 2016], it is still a challenging task to acquire such tool information during surgeries. This information is commonly acquired whether by manual annotation or by using an additional equipment to capture the data, which renders the approach impractical. In this work, we tackle the automatic surgical tool recognition using only vision-based methods in order to preclude the need of human intervention to get the tool usage signals. But, it is actually a challenging task to recognize the tools in the microscope images because only the tools tips are visible in the microscope field of view. In addition, these tools have a wide variety of shapes and some of them have very similar shapes. So, we hypothesized that finding discriminative features for differentiating between the tools is challenging if we would only work with the surgery videos issued from the microscope. To bypass the aforementioned challenges, we propose in this thesis the addition of a second video stream, filming the surgical tray. The surgical tray holds the tools and the supplies that are expected to be required to complete the surgical procedure. In contrast to the microscope, it allows the tools to be laid out and displayed in an easily recognizable fashion, as shown in the Fig.2.7b. This is expectedly taming the complexity of the problem. However, it is necessary to note that the surgical tray may hold non-metallic materials and the surgical tools wraps/packages along with a variety of preliminary actions that can be done by the surgeons. The constraints and the challenges existing in the surgical tray are described in details in the section 3.3.3.
In practice, the surgical tray allows the surgeons and their assistants to retrieve the correct tool without delay. A tool used by the surgeons in the eye (present in the tool-tissue interaction videos) means that this tool has been taken out first from the surgical tray few seconds before. In addition, a tool put on the surgical tray by the surgeons means that this tool has left the eye (not present any more in the tool-tissue interaction videos) few seconds before. Therefore, the general rule is: by



(a) Microscope field of view

(b) Surgical tray

Figure 2.7: Surgical tray image captured at time t. Microscope image captured at time t + a few seconds, showing part of the knife that has been taken out from the tray.

knowing **which tools are put on or taken from the surgical tray we know which tools are likely being used by the surgeon and which tools surely are not**, as illustrated in the Fig.2.7. In this thesis, we perform surgical tool recognition on two video types: tool-tissue interaction videos and surgical tray videos. Our main aim is to recognize the tools in the microscope videos due to their discriminant properties in any SAR system. Indeed, we believe that the surgical tray tool information can be a complementary element to the surgical tools information in the tool-tissue interaction videos. Ideally, this should boost the performance of any automatic surgical tool recognition system on the tool-tissue interaction videos.

In previous sections, we have detailed the existing approaches proposed to tackle these two intertwined fields: SAR and surgical tool recognition. Beside the limitations of the results in the SAR methods, these methodologies have demonstrated that the results obtained in the surgical tool recognition methods are encouraging [Mishra et al., 2017] [Hu et al., 2017] even though there are only few public datasets available. In this rundown, we present a thorough surgical tool recognition method started by generating a dataset containing recording of real cataract surgeries where each surgery is recorded in two videos: one for the surgical tray and the other one for the tool-tissue interaction. Thereafter, two different approaches were followed to detect the tool presence in both videos: one is a patch-based solution with traditional handcrafted features or learning features and the other one identical to the latest trends in the surgical tool recognition field [Mishra et al., 2017] [Hu et al., 2017] [Twinanda et al., 2016].

In the patch-based approach, we construct a pipeline consisting of: (1) the extraction of visual features [Blum et al., 2010] [Dergachyova et al., 2016] [Twinanda et al., 2015] [Lalys et al., 2012], such as color information and intensity gradients; (2) the usage of traditional classification frameworks, such as k-nearest neighbours.

Various studies have tackled the detection of surgical tools presence on *m2cai16-tool* dataset, resulting in new state-of-the-art performance. However, in our case, we

deal with a more challenging context. The number of tools to be detected is three times more than the number of tools used in *m2cai16-tool* dataset. In our tool-tissue interaction videos, only the tips of the tools are present and many of which resemble strongly. On the surgical tray, these tools have a wide diversity of shapes ranging from tiny tools (like canulas and needles) to surgical tool packages/wraps, consequently complicating the realization of a system capable of handling all these challenges at once. With the emergence of deep learning methods across all domains, the best performing CNN architectures on ImageNet dataset are used. To alleviate the inherent challenges of the surgical tray, we propose to work on simulated tray scenes along with a patch-based CNN approach in order to refine the performance of the system. Ultimately, in addition to the visual features extracted by the CNNs, the temporal information of both videos are used in a LSTM network to learn the temporal connectionism. The employment of temporal information in any surgical tool detection system can lead to better performance. In parallel, a very recent work [Mishra et al., 2017] proposed the same idea of exploiting the temporal information on *m2cai16-tool* dataset. In accordance with our proposition, they demonstrated that the temporal connectionism is a source of decreasing the local error in detecting the tool presence. To the best of our knowledge, this work and [Mishra et al., 2017] are the first few studies which incorporate spatial and temporal information in deep learning based methods to address the surgical tool recognition problem.

"Data is the new oil."

Sir Arthur Conan Doyle

3

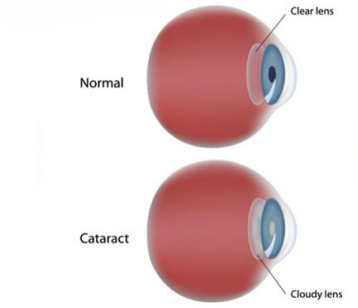
Cataract Surgery Data Description

Chapter Content

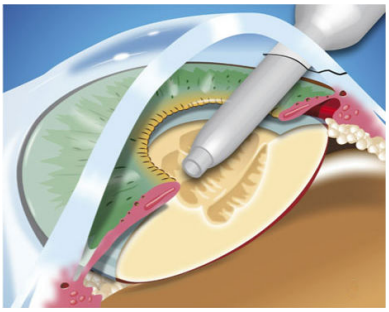
Due to the limited number of surgical tool recognition datasets and the importance of the surgical tray information, we generate in this thesis our own dataset of cataract surgeries videos, thanks to a collaboration with the ophthalmology department of the CHRU of Brest. We called it a real-world (RW) dataset. In the following section, we describe the cataract surgery procedure. Then, we present the dataset built to evaluate the methods proposed in this thesis.

3.1 Cataract Surgery

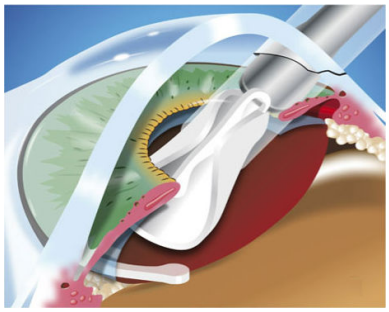
Cataract is a disease caused by a progressive opacification of the natural lens of the eye. The lens is normally transparent and its opacification causes a decrease in vision more or less important, occasionally leading to a vision loss (see Fig.3.1a). Cataracts are prevalent among older adults. In France, there are approximately 600,000 interventions each year[1]. It affects one in five after 65 years of age and one in three after 75 years of age and two out of three after 85 years of age. Indeed, cataract surgery is the most common surgical procedure worldwide: nineteen million cataract surgeries are performed annually [Trikha et al., 2013].



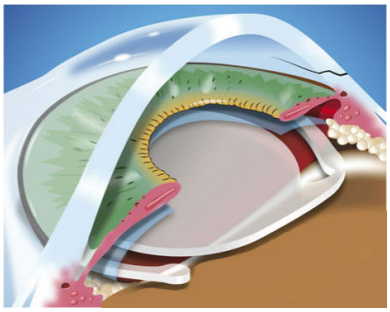
(a) Difference between clear and cloudy lens



(b) The cloudy lens is removed (phacoemulsification)



(c) A clear artificial lens is implanted into the empty capsule of the lens



(d) Intraocular lens (IOL) in place.

Figure 3.1: Main phases in the cataract surgery procedure. These images are a modified version of images got from this site[2].

When this opacification can not be corrected with glasses, contact lenses or corneal refractive surgery like LASIK, the only treatment is the cataract surgery. As illustrated in Fig.3.1, the procedure consists of removing the cloudy lens and to replace it by an artificial implant, called an intraocular lens (IOL) to restore the clear vision. In fact, two small incisions are made to reach the lens (primary and secondary incisions). Then, an opening in the capsule of the lens is made to break up the cloudy lens into small pieces. These pieces are then gently removed from the eye with irrigation/aspiration. The emulsification phase is called "Phacoemulsification" (see Fig.3.1b). Afterwards, an IOL is implanted into the empty lens capsule. Modern

[1] https://www.ouest-france.fr/leditiondusoir/data/9900/reader/reader.html#!preferred/1/package/9900/pub/14044/page/2

[2] http://www.ranelle.com/cataract-surgery/

cataract surgeries have made incision wounds self-sealing. There remain a few cases, however, that require placement of sutures in order to ensure wound safety.

The procedure is typically performed under local anesthesia and on an outpatient. The operation lasts on average 15 minutes. A surgical microscope is used by the surgeon to perform the procedure. This microscope contains a camera with a recording system linked to a screen in the OR that shows in real-time the operative field. Cataract surgery is widely practiced and it is being video-monitored during the last few years. A more complicated surgery would require a massive amount of data to cover all possible cases. Therefore, the cataract surgery can be deemed as an ideal field of study that allows the development and evaluation of computer-assisted surgery systems.

3.2 Video Acquisition

In order to amass a new dataset of videos of the cataract surgery, I visited once a week (one to seven hours), during a period of nine months, the ophthalmology department ORs of the CHRU of Brest. In contrast to the microscope, which has a built-in camera with a recording system, the surgical tray is not outfitted by a video acquisition system. Also, altering the surgical procedure is not commonly feasible for the clinical staff even though, for installing another camera in the OR, the surgical procedure is not meant to be altered per se. Then, the main challenge was the installation of another camera that shoots solely the surgical tray. The complexity lied in the ability to shoot a clear and complete view of the surgical tray. This requires to install the camera in a still position and as much closer as possible to the surgical tray. In fact, this is not acceptable from a clinical point of view unless the camera is sterilized, cleaned and covered appropriately. Starting from these conditions, the purpose of our first few visits was to coordinate with the clinical staff (managers and scrub nurses) to find the easiest and safest solution for installing the second camera in the OR. In the following sections, we describe our observations during the common three phases of the cataract surgery: pre-, intra-, and postoperative phases.

3.2.1 Preoperative Phase

At the entrance, a surgical attire is required to enter the ORs. It consists of scrub suits, cap/hoods, masks and gloves. Inside the ORs, the scrub nurses are responsible for cleaning and sterilizing everything surrounding the surgeons before the surgery and for preparing the patients appropriately for surgery, for instance putting dilating eye drops to enlarge the pupil of the eye. They are also in charge of preparing the surgical tray where they arrange/prep the surgical tools over the tray in order to be easily accessible by the surgeons during the surgery. Once the preparation step is done, the surgeons start the surgery. In Brest University Hospital, the surgeons use the OPMI Lumera T microscope (Carl Zeiss Meditec, Jena, Germany), illustrated in Fig. 3.2, to magnify the surgical field which helps in performing the surgical tasks better (accurate incisions etc.). This microscope has a built-in 180I camera (Toshiba, Tokyo, Japan) and linked to a MediCap USB200 recorder (MediCapture,

Plymouth Meeting, USA). A simple push to a button starts the recording of the operative field (the tool-tissue interaction videos).

However, it is forbidden to fix the camera, as it is, on the surgical tray for asepsis reasons. In addition, sterilizing the camera was not a feasible task. Furthermore, unless you are one of the surgeons, no one is allowed to touch the surgical tray from the moment they start preparing it until the end of the surgery. After a few tries of shooting a surgery using a camera held by one hand, an articulated arm was fixed using a clamp on the surgical tray.

(e) OPMI Lumera T microscope

(f) OPMI Lumera T microscope ready to use in ocular surgeries

Figure 3.2: OPMI Lumera T microscope for ophthalmic surgeries.

Figure 3.3: The camera fixed on the surgical tray in the OR.

This arm was covered by a surgical tray drape. Ultimately, the Sony HDR-

PJ5301[3] video camera was attached to this arm and covered by a plastic bag punctured in front of the camera lens, as illustrated in Fig.3.3. To start the recording of the surgical tray scene, a push to a button on the camera is required. This is different than the button dedicated for the operative field recordings.

3.2.2 Intraoperative Phase

The intra-operative period begins when the surgeon starts the intervention. During a normal conduct of the surgery, a succession of surgical steps is done by the surgeons in the surgical field (incisions, hydrodissection, phacoemulsification, IOL implant and suture if needed). In contrast, the surgeon and the intern can do surgery related and non-surgery related tasks over the surgical tray. For instance, they prep the IOL for implantation or they can fill the syringes. However, they can just arrange the tools over the tray or taking apart cannulas from syringes as a preliminary step for cleaning in after surgery. The surgical tray is then deemed as a work desk. In addition, it is not always necessary to have all the tools presented on the surgical tray before starting the surgery. Thus, new tools can be dropped on the tray during the surgery depending on the state of the patient. Those tools can be put on first on the surgical tray and then used in the operative field or vice-versa.

3.2.3 Postoperative Phase

As the surgeon finishes the surgery, two buttons are pushed to stop the recording of both scenes (the surgical field and the surgical tray). A shield is placed over the eye to protect it in the early stages while the patient heals from surgery. At this moment, the scrub nurses discard the disposable surgical tools, however, the non-disposable tools are packaged and sent for cleaning and sterilizing in order to be used later in another surgery. Therefore, we were able to keep a collection of the disposable tools used during the surgery.

3.3 Description

In this section, we present in details the specification of the dataset prepared in this thesis. First, we introduce the surgical tools commonly used in any cataract surgery procedure. Then, we describe the videos shot in the OR for the operative field and the surgical tray (duration, resolution etc.).

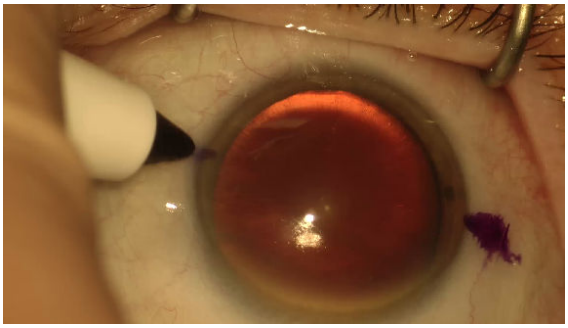
3.3.1 Tools

All surgical tools visible in the tool-tissue interaction videos were listed and labeled by the surgeons. It consists of a collection of 21 surgical tools: 12 disposable tools and 9 non-disposable tools. Each surgical tool has one or multiple roles in the surgery procedure. The roles of each surgical tool are reported in Table.3.1. The two complementary views of each surgical tool are illustrated in Fig.3.5.

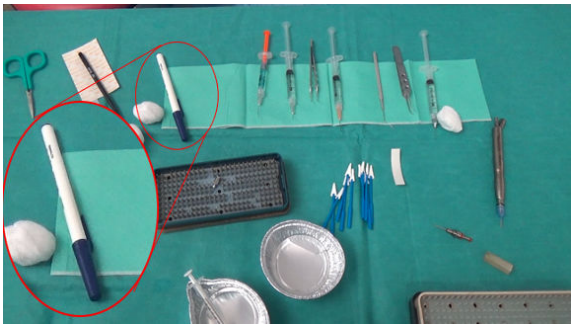
[3] http://www.magazinevideo.com/fiche-technique/sony-hdr-pj530/29798.htm

Tool	Role
biomarker	used in some specific cases where marks are required to identify how to put in place the IOL.
Charleux canula	aspires the residual masses after phacoemulsification.
hydrodissection canula	separates the anterior capsule from cortex by jet of water.
Rycroft canula	injects a balanced salt solution (BSS) into the corneal stroma until diffuse whitening is observed.
viscoelastic canula	injects a dispersive viscoelastic substance to coat and protect the corneal endothelial cells.
cotton	absorbs the excess fluid.
capsulorhexis cystotome	ruptures the anterior face of the capsule of the lens.
Bonn forceps	holds the sclera while doing the main incision.
capsulorhexis forceps	removes the anterior face of the capsule torn by the capsulorhexis cystotome.
Troutman forceps	used in the surgical field often to hold the suture needle. However, on the surgical tray, it is usually used to prepare the IOL.
needle holder	holds the needle when a surgical stitching is required.
irrigation/aspiration handpiece	aspires the lens cortex and polishes the capsule.
phacoemulsifier handpiece	sculpts and emulsifies the lens while aspiring particles through the tip.
vitrectomy handpiece	removes the vitreous in case of capsular rupture.
implant injector	implants the IOL into the capsule of the lens.
primary incision knife	incises the bottom of the cornea. This incision is deemed as an entry point for the main tools used in the surgery.
secondary incision knife	incises the left part of the cornea in order to let some tools help in manipulating the lens.
micromanipulator	allows to manipulate the lens and to maintain the fragments in the phacoemulsification phase.
suture needle	It is the needle used to do the suture step.
Mendez ring	It is a protractor used in measuring the insertion angle of the IOL.
Vannas scissors	cuts the suture thread.

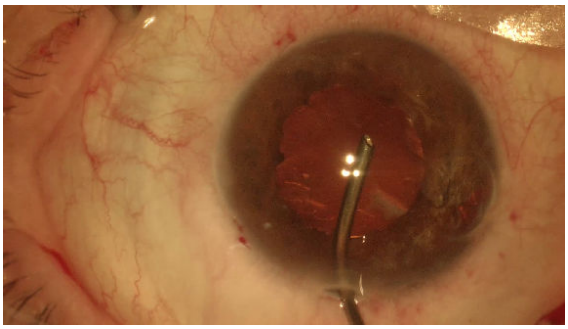
Table 3.1: The surgical tools commonly used in the cataract surgery and their roles in the surgery procedure. Disposable surgical tools are in bold.



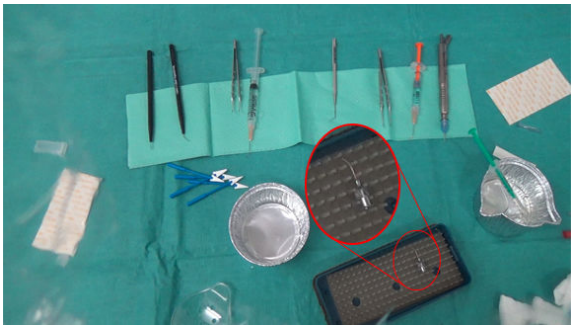
(a) biomarker



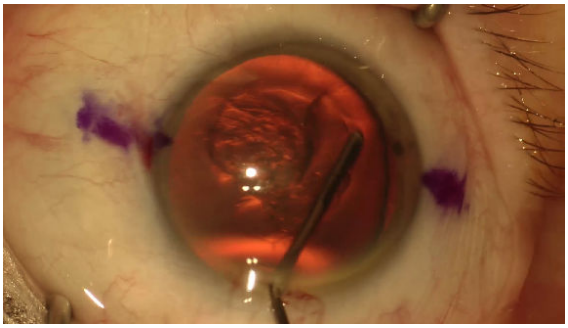
(b) biomarker on the tray



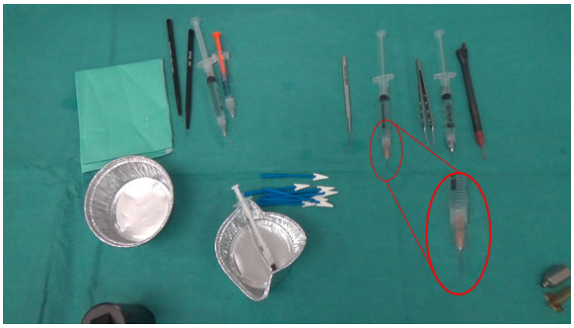
(c) Charleux canula



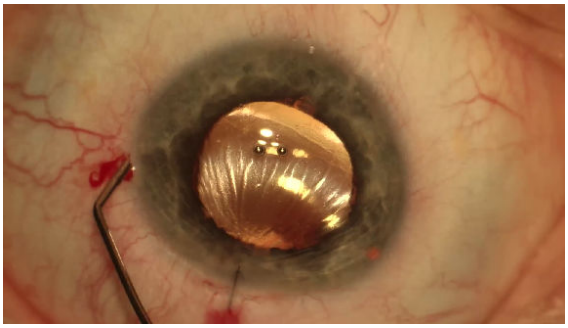
(d) Charleux canula on the tray



(e) hydrodissection canula



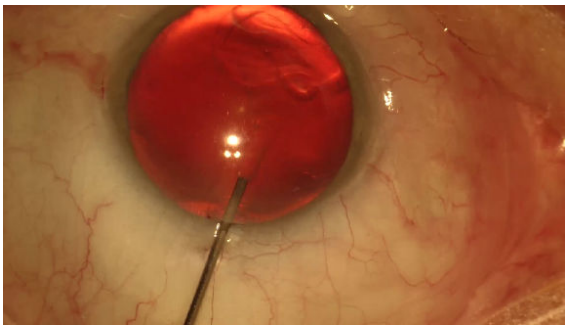
(f) hydrodissection canula on the tray



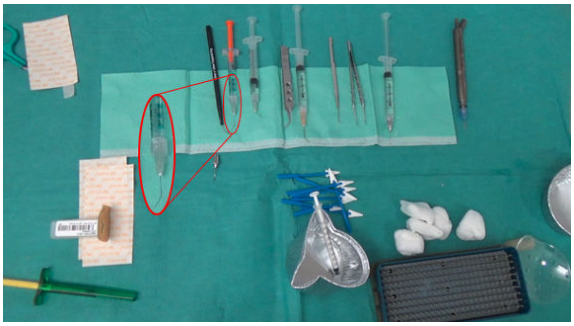
(g) Rycroft canula

(h) Rycroft canula on the tray

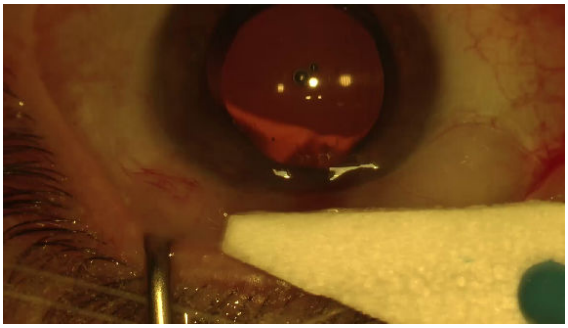
Figure 3.4: The surgical tools annotated in the tool-tissue interaction videos and their full-view version on the tray.



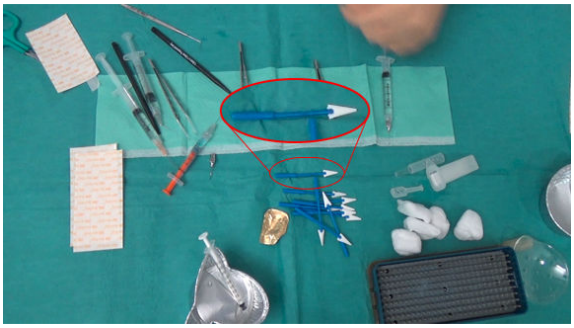
(a) viscoelastic canula



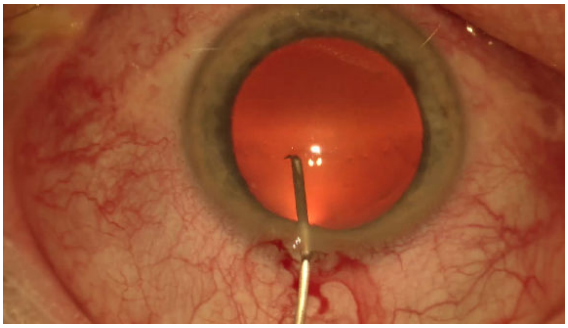
(b) viscoelastic canula on the tray



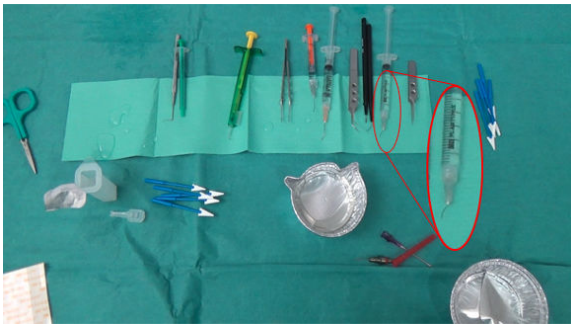
(c) cotton



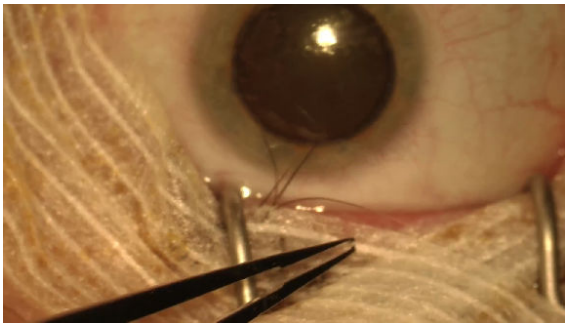
(d) cotton on the tray



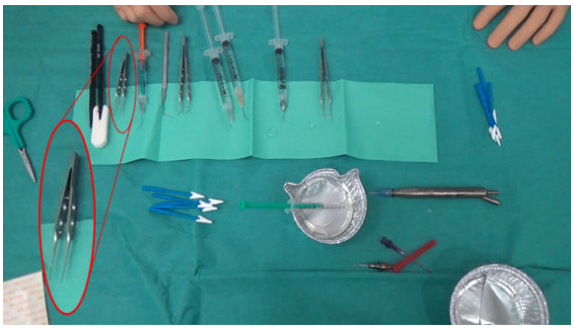
(e) capsulorhexis cystotome



(f) capsulorhexis cystotome on the tray

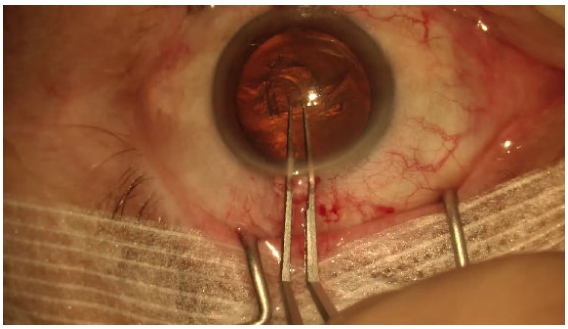


(g) Bonn forceps

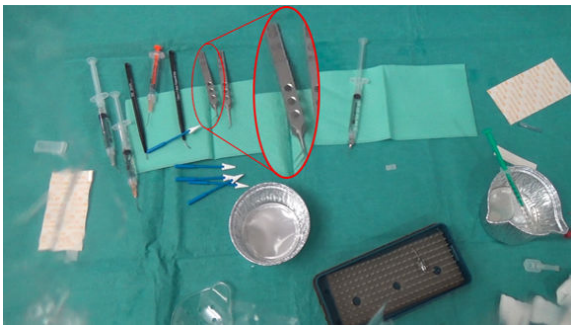


(h) Bonn forceps on the tray

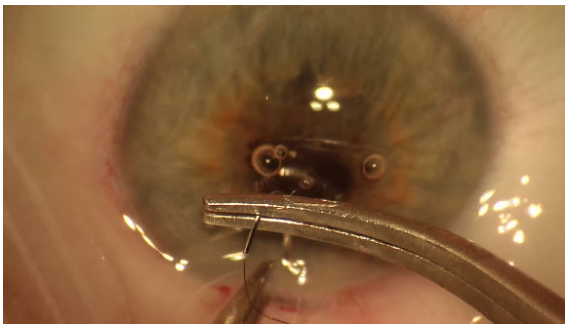
Figure 3.5: Figure 3.4 (Cont.).



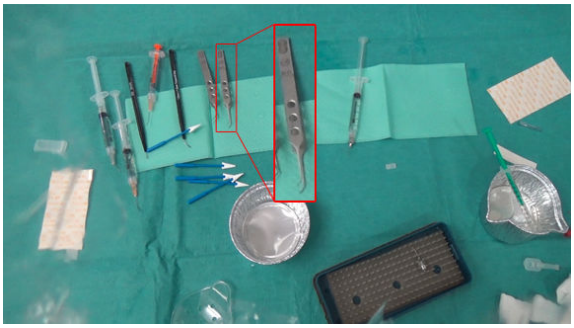
(a) capsulorhexis forceps



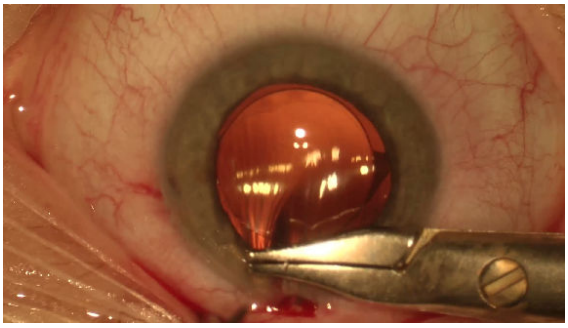
(b) capsulorhexis forceps on the tray



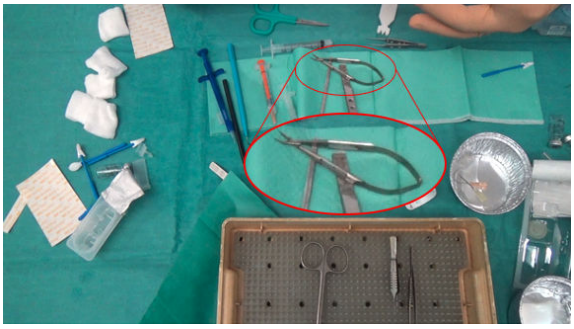
(c) Troutman forceps



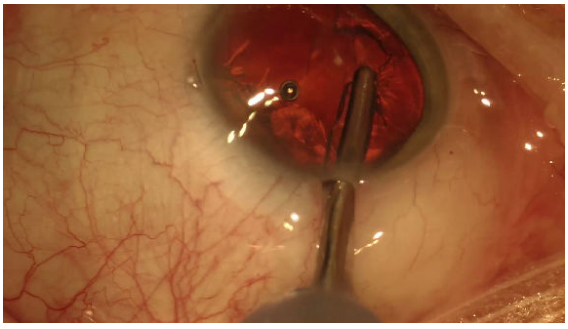
(d) Troutman forceps on the tray



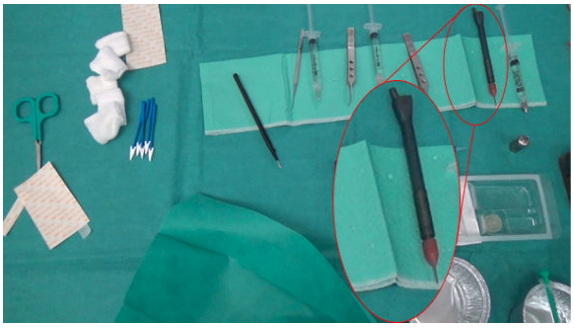
(e) needle holder



(f) needle holder on the tray

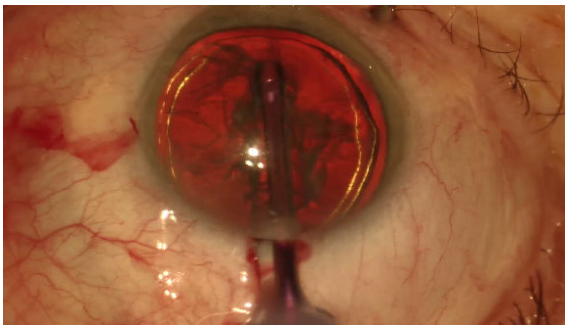


(g) irrigation / aspiration handpiece

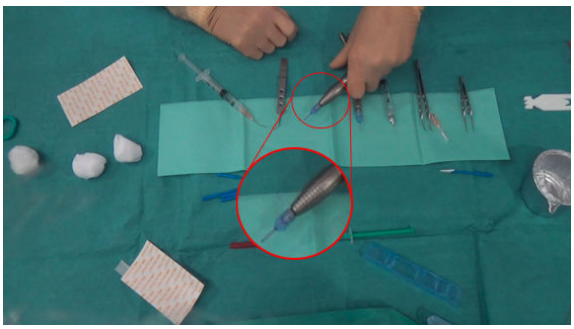


(h) irrigation / aspiration handpiece on the tray

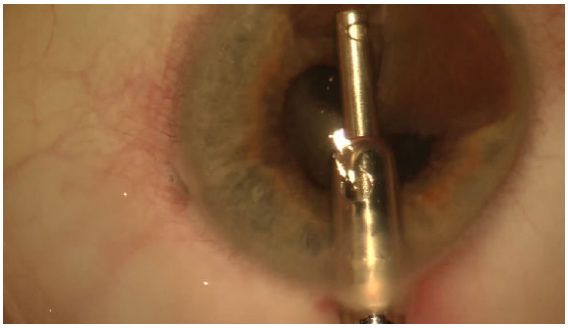
Figure 3.6: Figure 3.5 (Cont.).



(a) phacoemulsifier handpiece

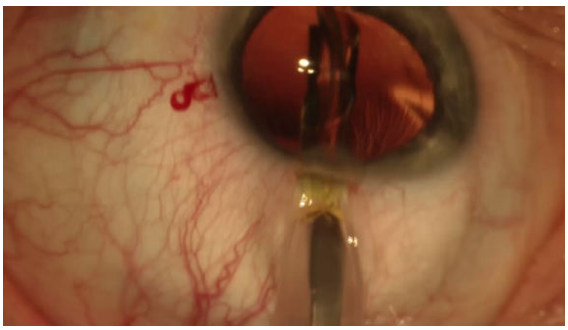


(b) phacoemulsifier handpiece on the tray



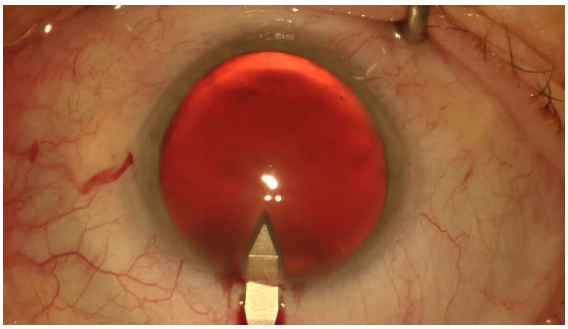
(c) vitrectomy handpiece

(d) vitrectomy handpiece on the tray



(e) implant injector

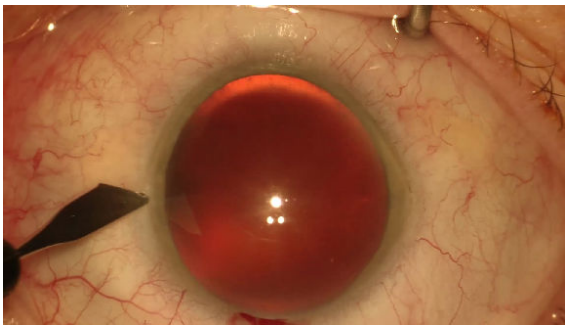
(f) implant injector on the tray



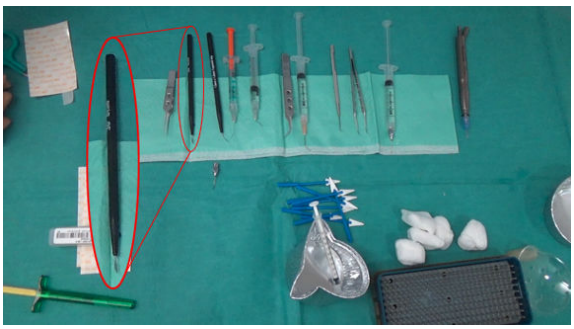
(g) primary incision knife

(h) primary incision knife on the tray

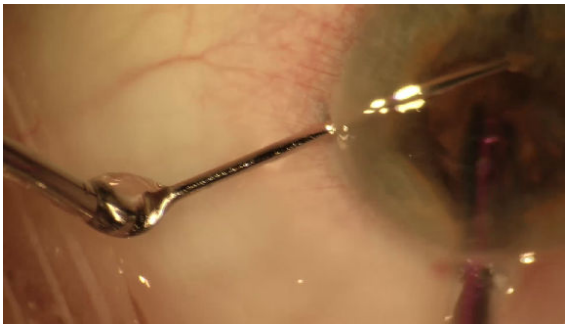
Figure 3.7: Figure 3.6 (Cont.).



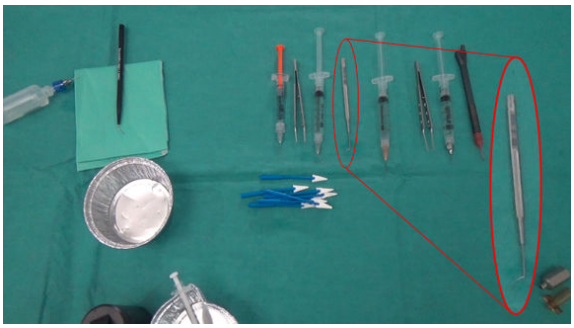
(a) secondary incision knife



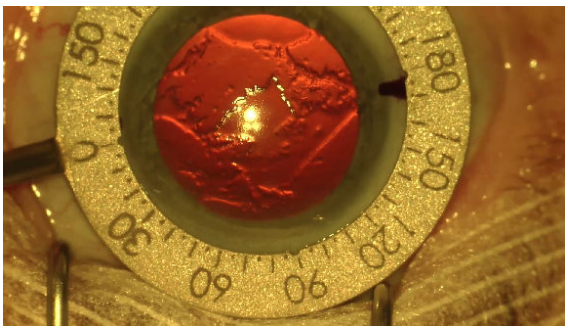
(b) secondary incision knife on the tray



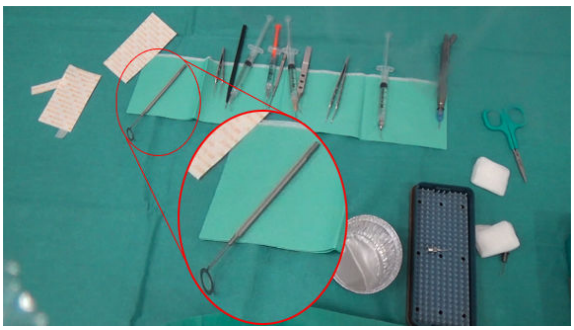
(c) micromanipulator



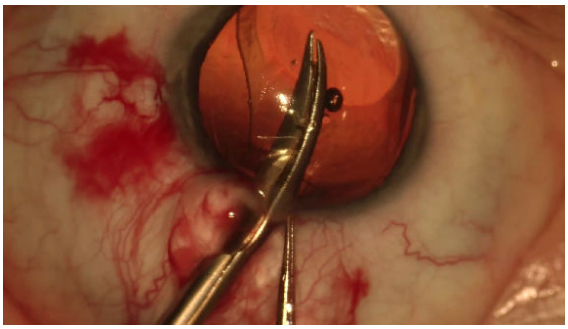
(d) micromanipulator on the tray



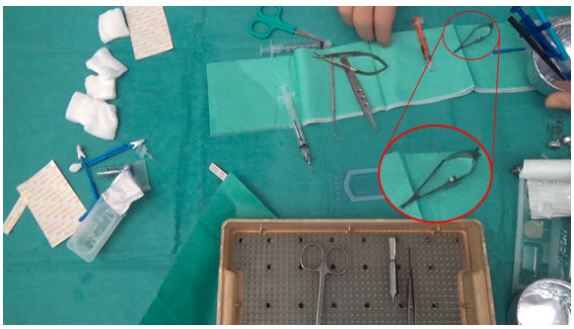
(e) Mendez ring



(f) Mendez ring on the tray

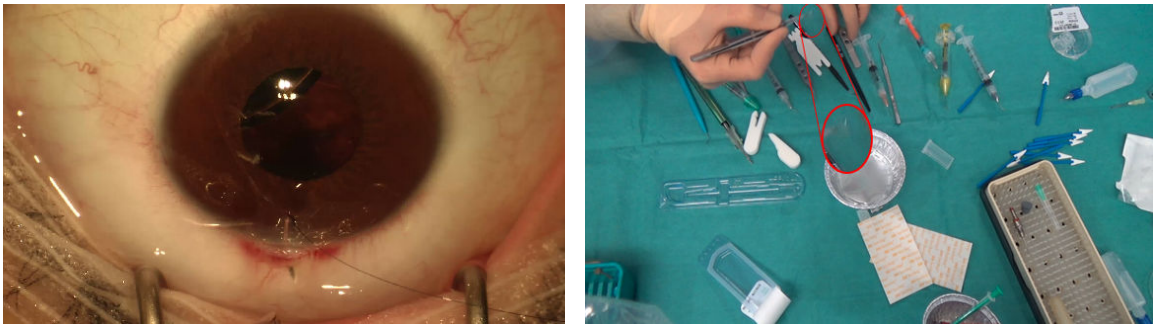


(g) Vannas scissors



(h) Vannas scissors on the tray

Figure 3.8: Figure 3.7 (Cont.).



(a) suture needle (b) suture needle on the tray

Figure 3.9: Figure 3.8 (Cont.).

3.3.2 Videos

The dataset consists of 50 videos of cataract surgeries performed in Brest University Hospital. Patients were 61 years old on average (minimum: 23, maximum: 83, standard deviation: 10). Surgeries were performed by three surgeons: a renowned expert (48 surgeries), a one-year experienced surgeon (1 surgery) and an intern (1 surgery). Each surgery was recorded in two videos: the tool-tissue interaction video and the surgical tray video. The frame definition was 1920x1080 pixels (full HD resolution) for both types of videos. The frame rate was approximately 30 frames per second for the tool-tissue interaction videos and 50 frames per second for the surgical tray videos. Microscope videos had a duration of 10 minutes and 56 s on average (minimum: 6 minutes 23 s, maximum: 40 minutes 34 s, standard deviation: 6 minutes 5 s). Surgical tray videos had a duration of 11 minutes and 3 s on average (minimum: 6 minutes 30 s, maximum: 40 minutes 48 s, standard deviation: 6 minutes 3 s). In total, more than nine hours of surgery (for each video type) have been video recorded.

3.3.3 Constraints and Challenges

Ideally, detecting the surgical tools over the tray is simpler than detecting them in the surgical field because of the way the tools are laid down on the tray. However, the surgical tray may contain tools/objects that are repeatedly unused. Also, it is the place where the surgeons prep their preliminary actions before accomplishing any task in the surgical field. Thus, analyzing the surgical tray videos is also challenging due to the specification of the surgical tray and to the variety of actions that can be realized by the surgeons on it, such as preparing implant, filling in the syringes, etc.

Initially, the surgical tray camera does not have a full view of the surgical tray due its position relative to the tray. This may lead to have partially visible tools or tools completely out of the camera field of view whereas they are present on the tray. The lighting conditions of the OR are not best fitted for our recording setups on the tray. This may produce noises in the images captured by the camera. Also, there is no possibility to start and stop the two recording systems (microscope and camera) at the same time, expectedly leading to have an offset of few seconds between both

videos.

Numerous tools have low distinctive patterns, for instance the knifes, cannulas and forceps. At the outset of the surgery, the tools required to finish the surgery procedure are present on the tray and their order is subject to change from surgery to another. However, during the surgical procedure, new tools are often put on the surgical tray. This may be due to a complication in the surgery or the surgical workflow followed by the clinical team or tools missed when preparing the tray. Some objects/tools on the tray are not meant to be used in the surgical field. In fact, they are dedicated for the surgical environment or to help accomplishing tasks over the tray per se (see Fig.3.10f). Another type of objects over the tray is the unused objects, such as the tool packages/wraps (see Fig.3.10c). During the surgery, when a tool is taken from the tray, the surgeons will put it back up to tens of seconds later (the assistant can keep the tool in his/her hands to make it easier for the surgeon to move faster in the surgery). There are some exceptions to this rule where the tool put on the tray can be out of the camera field of view or it is the *phacoemulsifier handpiece*. This tool is held by the phacoemulsification machine (see Fig.3.10d) and it rarely passes in the field of view of the camera fixed on the tray (see Fig.3.7a).

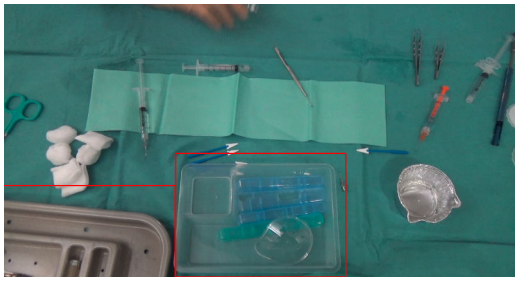
Additionally, gloved hands appear in the camera field of view to accomplish some actions: (1) take out or put a tool on the tray. (2) prepare tools for usage in the surgical field. (3) arrange the tools over the tray. Occasionally, the assistant can put his/her hand on the tray, leading to hide some tools for a certain period (see Fig.3.10a). Also, the whole tray is regularly moved due to the interaction with the surgeons. This produces a shakiness moments in the video and lead occasionally the plastic bag to move and to appear in the camera field of view (see Fig.3.10e). This last issue is presented in 14 out of 50 videos.

3.4 Ground Truth

The usage of each surgical tool in the videos was annotated independently by two non-M.D. experts. For the tool-tissue interaction videos, a tool was considered to be in use whenever it was in contact with the eyeball. Therefore, a timestamp was recorded by both experts whenever a surgical tool came into contact with the eyeball, and also when it stopped touching the eyeball. Up to three tools may be used simultaneously: two by the surgeon (one per hand) and sometimes one by an assistant. As for the surgical tray videos, a timestamp was recorded by both experts whenever a tool is put on the surgical tray, and also when it is taken from the surgical tray. A tool was deemed present on the tray whenever a part of it starts to appear in the field of view of the camera, whereas, it is considered vanished whenever the tool is completely out of the camera field of view. All surgery related tools/objects are present on the tray, leading to a significant number of objects present at once on the surgical tray. Additionally, some of the tools have more than one instance on the tray, such as the *cotton*. Then, the annotations were based on the number of instance of the tool present in the camera field of view for both video types, where zero indicates a vanished tool. Thereafter, annotations from both experts were adjudicated: whenever expert 1 annotated that tool A was being used, while expert 2 annotated that tool B was being used instead of A, experts watched the video

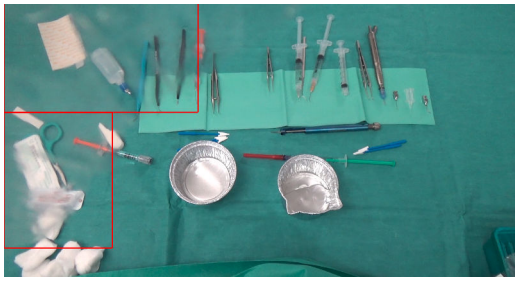
(a) Assistant's hand hidding some tools

(b) Obscure image

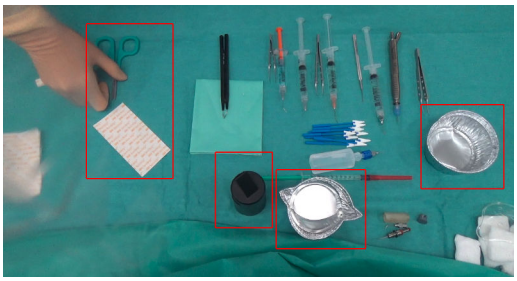


(c) Examples of surgical tool packages

(d) Phacoemulsifier handpiece held by the phacoemulsification machine



(e) Plastic bag blurs the field of view



(f) Examples of tools non-used in the surgical field

Figure 3.10: Various constraints and challenges on the surgical tray. Objects bounded box in red are the zones representing what is mentioned in the caption of each image.

together and jointly determined the actual tool usage. However, the precise timing of tool/eyeball contacts and tool appearing/disappearing was not adjudicated. We denote the number of instance of a tool annotated by expert 1 and 2 respectively by $nb\_instance_1$ and $nb\_instance_2$. Therefore, a probabilistic reference standard was obtained:

- $nb\_instance_1$: both experts agree on the number of instance of a tool
- $(nb\_instance_1 + nb\_instance_2)$ / 2 : experts disagree on the number of instance of a tool for the tool-tissue interaction videos.
- - $(nb\_instance_1 + nb\_instance_2)$ / 2 : experts disagree on the number of instance of a tool for the surgical tray videos. On the tray, more than one instance of a tool is **commonly** the case. The minus was added to differentiate between experts' disagreement and agreement.

The inter-rater agreement, before and after adjudication for both video types, is reported in Table 3.3 and Table 3.2.

Tool	Agreement before adjudication	Agreement after adjudication
biomarker	0.834	0.834
Charleux canula	0.949	0.963
hydrodissection canula	**0.868**	**0.982**
Rycroft canula	**0.881**	**0.918**
viscoelastic canula	**0.859**	**0.974**
cotton	0.946	0.946
capsulorhexis cystotome	0.994	0.995
Bonn forceps	0.792	0.797
capsulorhexis forceps	0.836	0.848
Troutman forceps	0.764	0.764
needle holder	0.630	0.630
irrigation/aspiration handpiece	0.995	0.995
phacoemulsifier handpiece	0.996	0.996
vitrectomy handpiece	0.998	0.998
implant injector	0.979	0.979
primary incision knife	0.958	0.961
secondary incision knife	0.846	0.852
micromanipulator	0.989	0.995
suture needle	0.893	0.893
Mendez ring	0.940	0.952
Vannas scissors	0.823	0.823

Table 3.2: Statistics about tool usage annotation in the tool-tissue interaction videos. The two columns indicate inter-rater agreement (Cohen's kappa) before and after adjudication; the largest changes are in bold.

Tool	Agreement before adjudication	Agreement after adjudication
biomarker	**0.836**	**0.998**
Charleux canula	**0.924**	**0.999**
hydrodissection canula	**0.654**	**0.997**
Rycroft canula	**0.931**	**0.997**
viscoelastic canula	**0.551**	**0.997**
cotton	**0.846**	**0.999**
capsulorhexis cystotome	**0.809**	**0.998**
Bonn forceps	**0.890**	**0.998**
capsulorhexis forceps	**0.393**	**0.996**
Troutman forceps	**0.711**	**0.997**
needle holder	0.818	0.818
irrigation/aspiration handpiece	**0.255**	**0.999**
phacoemulsifier handpiece	**0.000**	**0.986**
vitrectomy handpiece	0.971	0.990
implant injector	**0.935**	**0.998**
primary incision knife	**0.808**	**0.999**
secondary incision knife	**0.731**	**0.997**
micromanipulator	**0.889**	**0.999**
suture needle	**0.890**	**0.998**
Mendez ring	1	1
Vannas scissors	**0.942**	**0.997**

Table 3.3: Statistics about tool usage annotation in the surgical tray videos. The two columns indicate inter-rater agreement (Weighted Cohen's kappa) before and after adjudication; the largest changes are in bold.

Moreover, the annotations were performed at the frame level for both types of videos, using a web interface connected to a MySQL database (see Fig. 3.11). This web application contains a video interface to load the tool-tissue or the tray videos accompanied by the list of actions to be annotated and the 21 tools labeled by surgeons. When selecting a tool, a small text field appears where the number of instance of the tool can be filled in. Tool usage, during a typical surgery and a complicated one, is illustrated respectively in Fig.3.12-3.13 and Fig.3.14-3.15. Regardless of the exceptions, these images demonstrate the general concept followed in this thesis: telling which tool is put on or taken from the surgical tray (in other words which tools present or not present on the surgical tray), we can tell which tools are probably being used by the surgeons.

3.4.1 CATARACTS Challenge

In order to stimulate the research on automatic detection of surgical tool presence, the tool-tissue interaction videos of this dataset are released publicly in the context of

CATARACTS[4] (Challenge on Automatic Tool Annotation for cataRACT Surgery). It is organized to evaluate existing and new tool presence detection algorithms for the specific context of cataract surgery. It consists of detecting the presence of the 21 surgical tools described in section 3.3.1. It is not associated to any conference, thus, it was opened for a period of eight months and a journal paper (see Appendix C) describing and summarizing the top ranking methods was the outcome of this challenge. It was submitted lately to Medical Image Analysis. During this period, we had roughly 200 registered users where 14 participating teams have submitted their solutions. The number of submissions was between 1 to 6 submissions per team. The submitted solutions are primarily based on deep learning. Compared to other challenges on the same website, the number of registered users indicates admittedly the interest of the community in this topic and the huge number of submitted solutions can be considered as success.

[4] `https://cataracts.grand-challenge.org/home/`

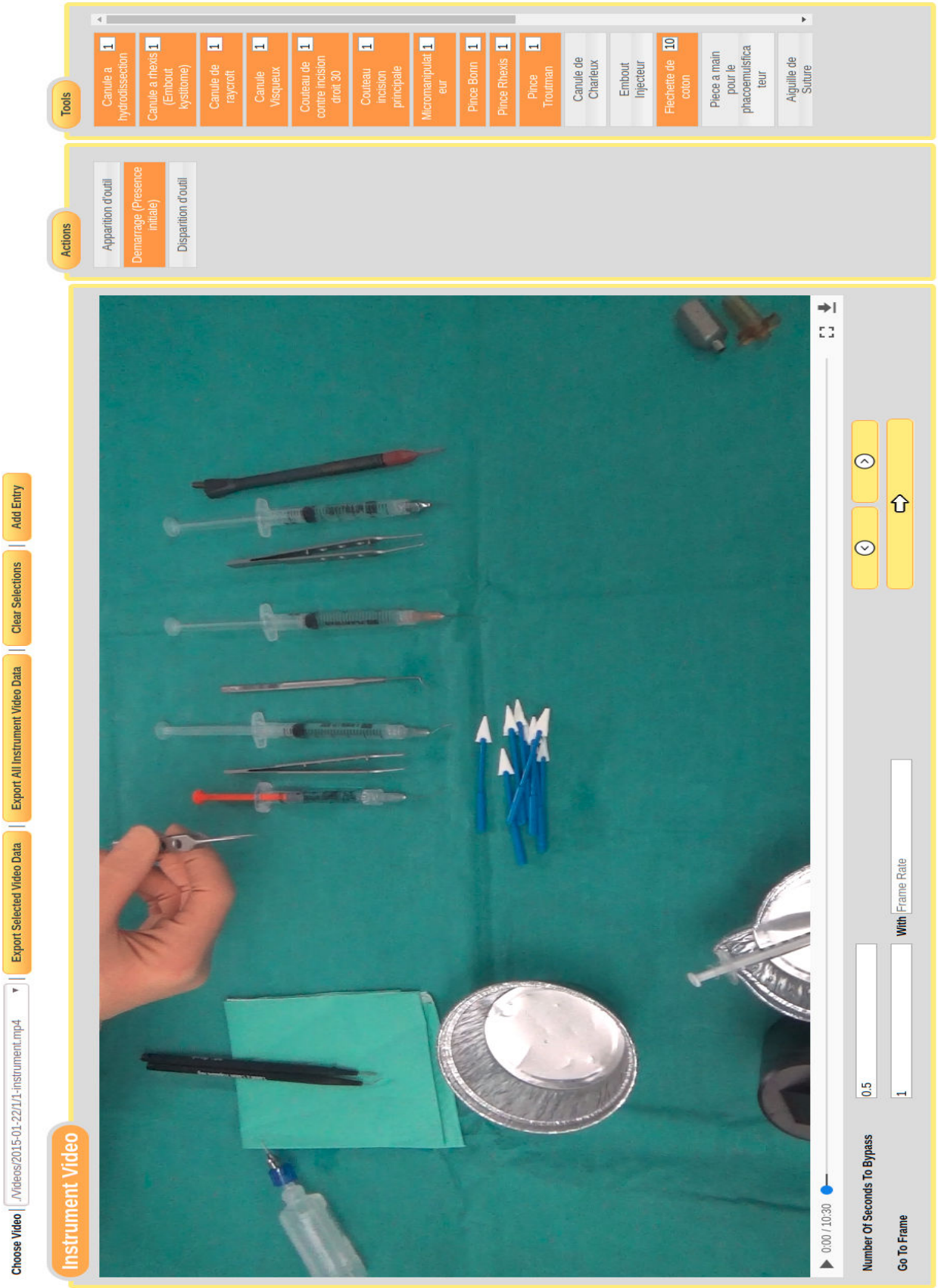


Figure 3.11: A web-based application for surgical tools annotation for the tool-tissue interaction videos and the surgical tray videos.

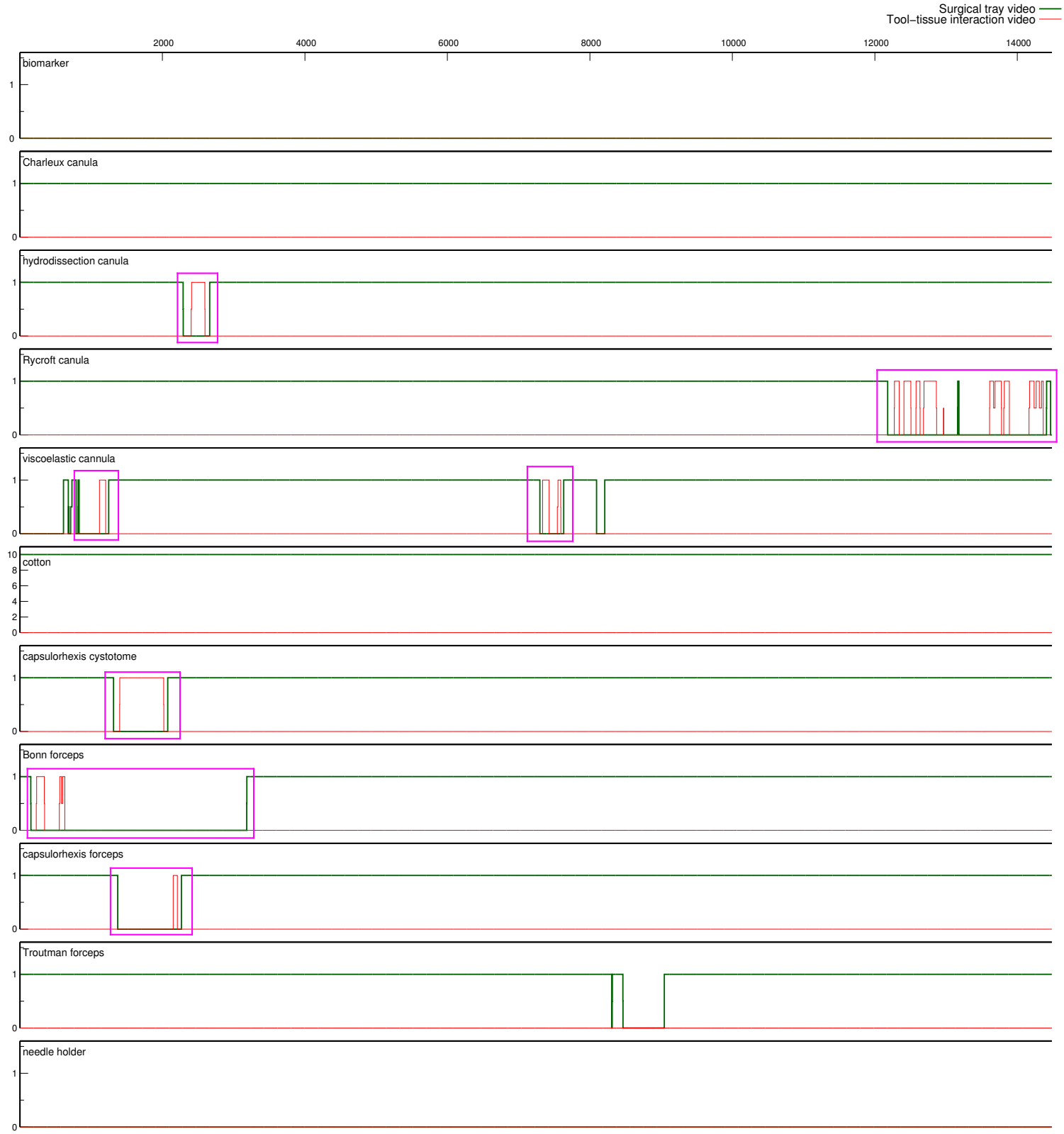


Figure 3.12: Tool usage during a typical surgery. Green and red indicates respectively the number of instance of tool present in each frame of the surgical tray video and the tool-tissue interaction video. Pink boxes indicate the moments where a tool is taken from the surgical tray, being used in the surgical field and probably put it back on the tray. Blue boxes show different types of exceptions.

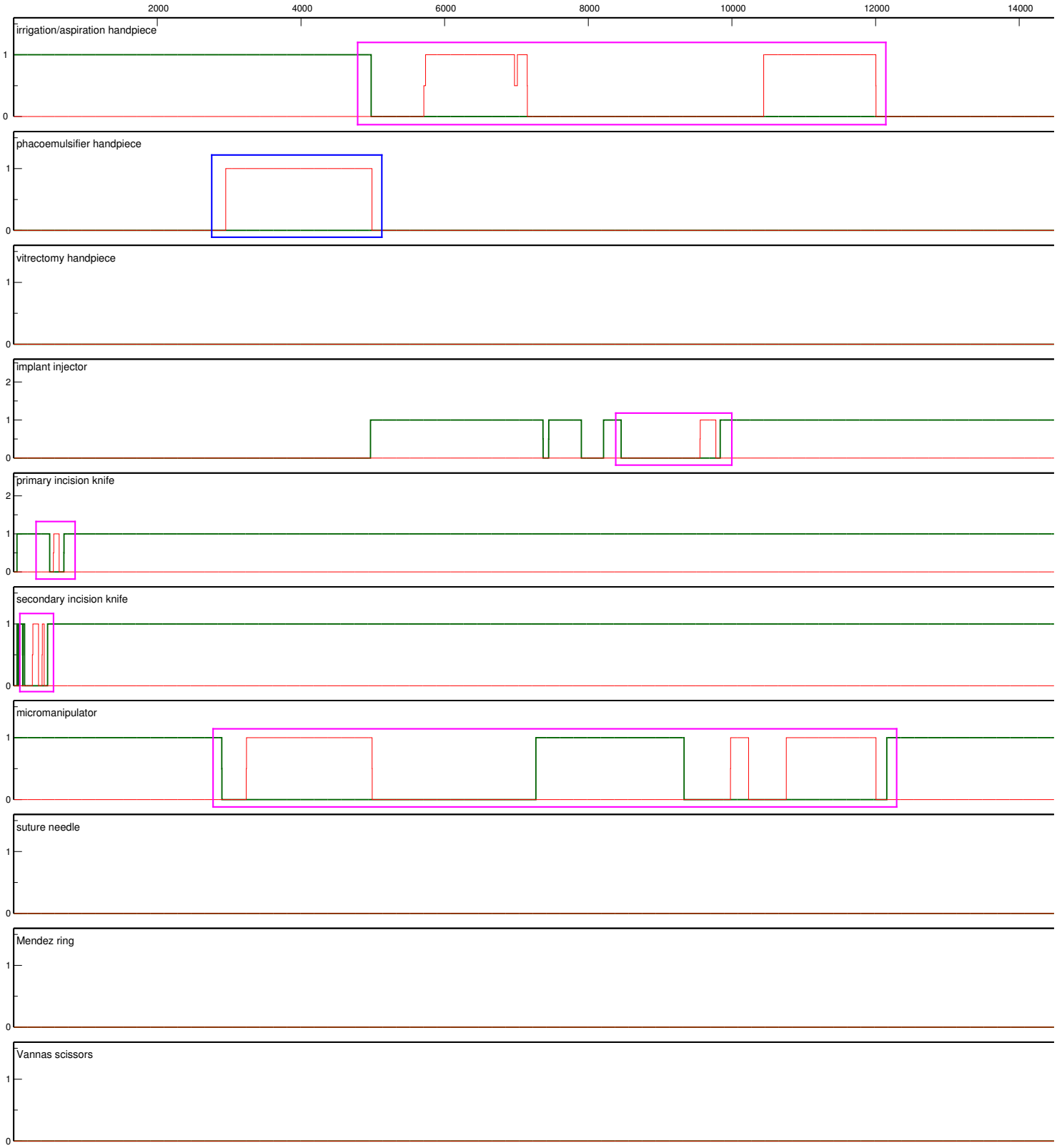


Figure 3.13: Figure 3.12 (Cont.).

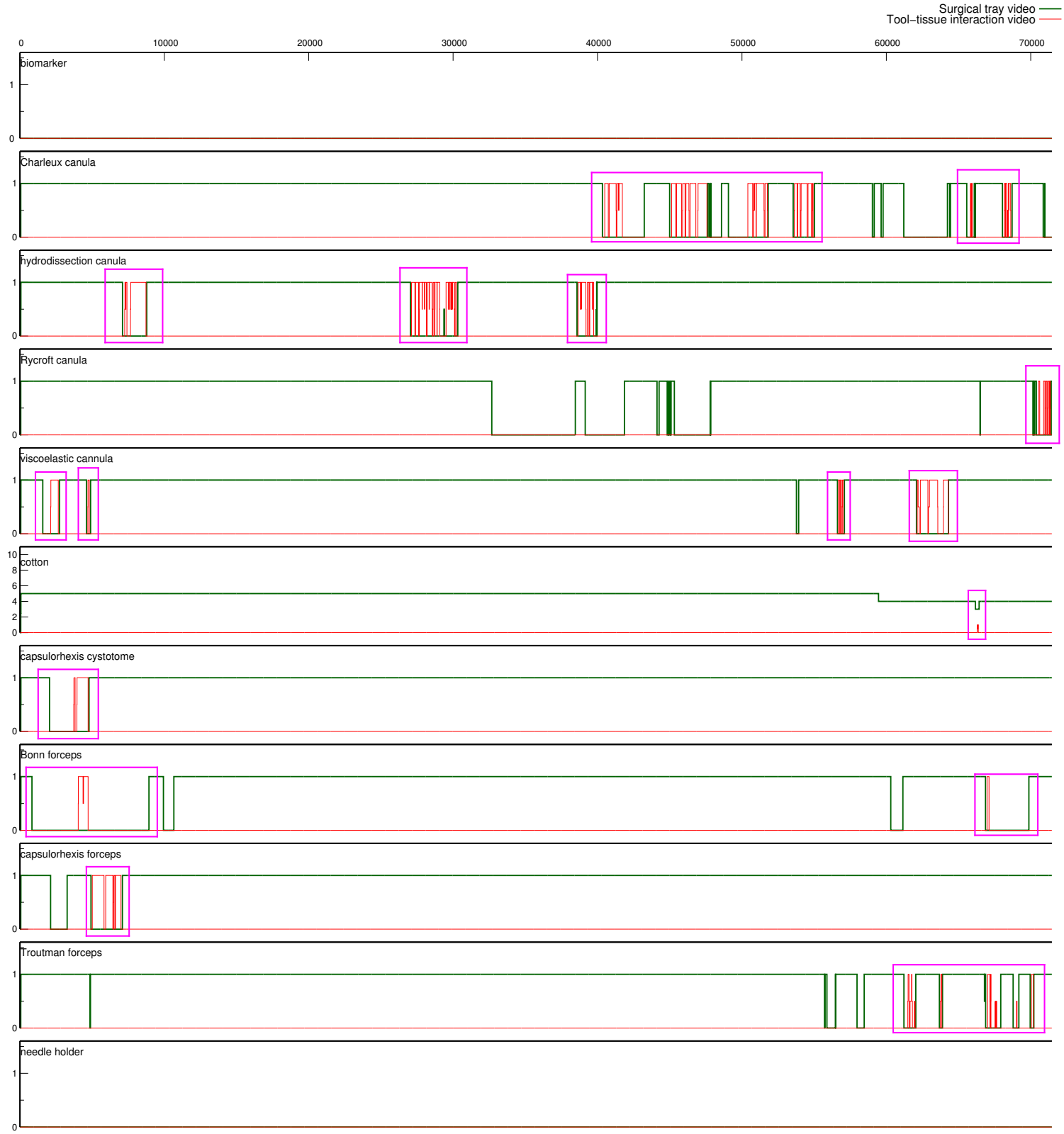


Figure 3.14: Tool usage during a complicated surgery. Green and red indicates respectively the number of instance of tool present in each frame of the surgical tray video and the tool-tissue interaction video. Pink boxes indicate the moments where a tool is taken from the surgical tray, being used in the surgical field and probably put it back on the tray. Blue boxes show different types of exceptions.

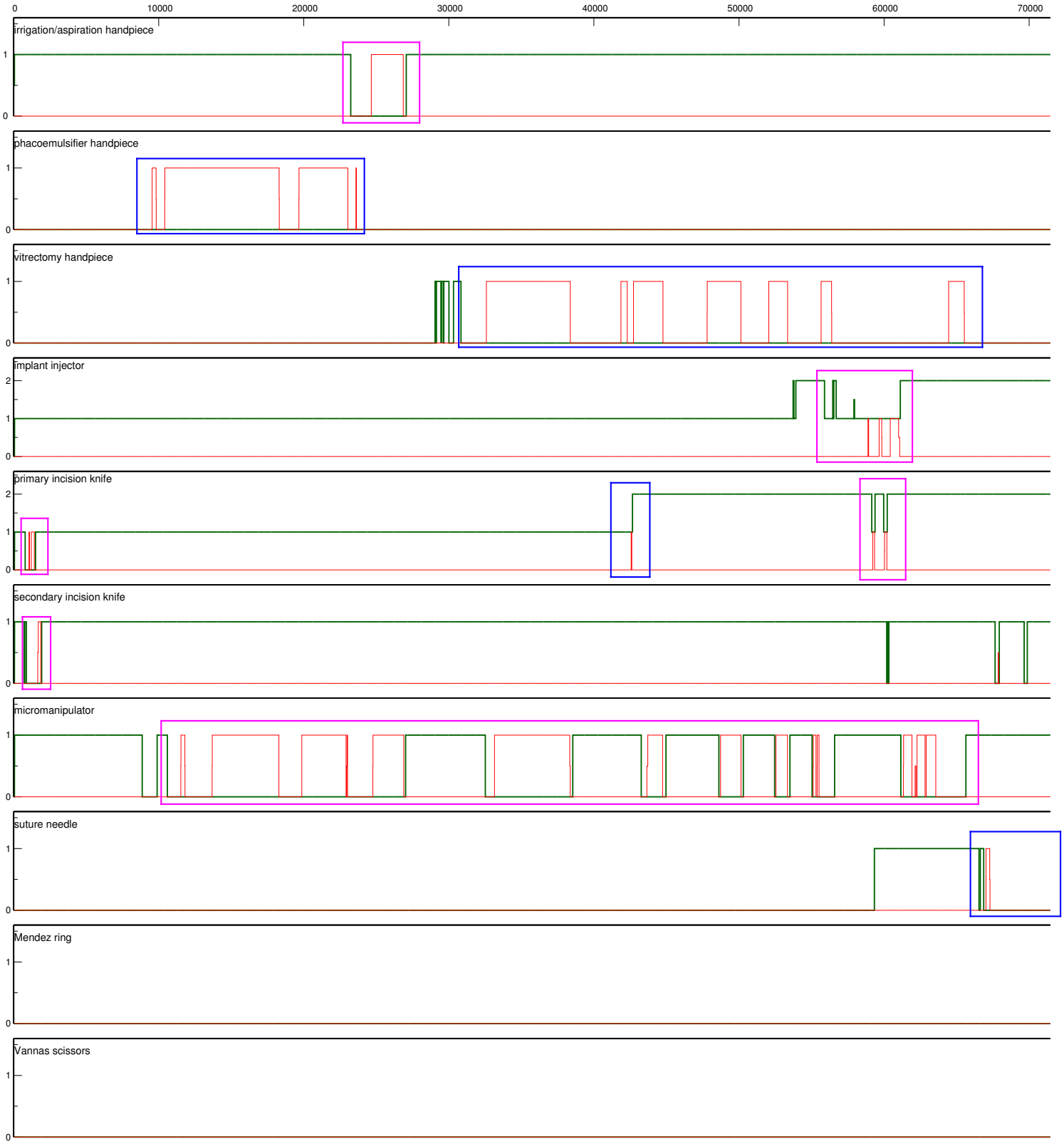


Figure 3.15: Figure 3.14 (Cont.).

"You don't learn to walk by following rules. You learn by doing, and by falling over."
Richard Branson

4

Surgical Tool Detection using Patch-based Approach

Chapter Content

As presented in the previous chapter, two videos are recorded along the surgery, one filming the surgical field and the other filming the surgical tray. The tools are present in both videos but with different appearances. At first glance, detecting the tools over the surgical tray is presumably deemed more straightforward than detecting the tools in the microscope field, because the tools are fully visible on

the surgical tray. In accordance with this hypothesis, various initial experimentations have been conducted in this thesis, to recognize the tools over the tray. In this chapter, we present a tool recognition pipeline using handcrafted visual features and shallow learning based features to address the surgical tool detection problem.
In order to know which tools are put on or taken from the surgical tray, two possible strategies can be followed: (1) **detect only the changes occurring along the surgery.** (2) **recognize at each instant of the surgery which tools are present on the tray.** The proposed solutions for both strategies are highly similar to one another. In fact, they are a local patch-based approaches, which have shown the capacity to cope with occlusions and to model the variability in object's shape as well as appearance [Teynor, 2006]. They consist of recognizing the tools in a local search windows instead of running the process over the whole image. In regards to the first strategy, the proposed pipeline is inspired from [Goyette et al., 2014], where they listed and compared numerous methods for detecting the changes in a video. The best approach presented was the block-matching approach built upon some traditional classification methods on top of some visual cues extracted from the image patches. Concerning the second strategy, the pipeline is a supervised patch-based approach applying a template matching technique. It consists of finding the reference patches of a tool in the real scene images. In addition, another template matching technique based on the *homography* transformation, detailed in Appendix A, has been tested to detect the tools over the surgical tray. In the following sections, we describe first the *change* detection where we try to detect the changes occurring on the surgical tray. Second, we present a method for detecting the surgical tool presence on the surgical tray videos.

4.1 Change Detection

This section concerns detecting the tools changes, consequently leading to tell, at each instant of the surgery, that a tool is probably present in the surgical field. To the best of our knowledge, this is the first study in the literature tackling the tools changes detection on surgical tray videos. Describing the changes can be considered a colossal opportunity to confront the intractable issues in the surgical tray scene, e.g. occlusion problems. During the surgery, there are various types of actions that can be done by the surgeons throughout. Simple and complicated tasks can be done by the surgeons. The surgeons do not simply put one tool on the tray and/or take one tool from the tray. In fact, the surgeons usually moves several tools around to search for the proper tool. In addition, some tools are used by the surgeons or the scrub-nurses to accomplish some tasks on the tray, e.g. preparing implants. Therefore, many tools are displaced without going out of the scene or used in the surgical field. It is worth mentioning that applying an optical flow solution is insufficient for such problems due to the large displacements involved and to the high similarities between tools. As illustrated in Fig. 5.12, the registration image is inadequate to isolate the targeted tool(s), thus the need for a thorough method to tackle this problem. Then, the main challenge is to differentiate tools that were simply moved around from tools that have put on or taken from the tray.

In the coming sections, we describe the pipeline used to perform the surgical tool

changes detection task. Then, we show the dataset used to evaluate this pipeline. Afterwards, we present the conclusions of the tools changes detection after presenting the experimental results.

4.1.1 Methodology

In this study, we apply a block matching approach by comparing the last image before an action is detected in the surgical tray scene, referred as I^l, to the first image after the action stops, referred as I^f. An action is deemed as an act of taking out a tool from the tray, putting it back, both at the same time or some other tasks done by the surgeons. The complete pipeline to perform the *change* detection is: (1) extracting features from each patch. (2) defining the change descriptor for the patches. (3) classifying the change descriptor into a tool or no tool.

4.1.1.1 Feature Extraction

Two types of features to represent the patches of the images were extracted for this method: handcrafted and learning features.

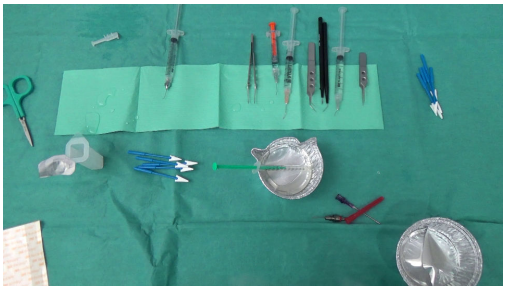
Handcrafted features. Simple visual features are proposed in this study. For each patch, *mean* and the *standard deviation* of the intensity values of the R, G, B, H, S and V channels were extracted as well as the *mean* and the *standard deviation* of the result of *Sobel* edge detection applied to the luminance channel. It results in a vector descriptor of 14 elements.

Shallow learning features. The second type of features is based on the principle component analysis (PCA) [Jolliffe, 2002]. PCA is an orthogonal linear transformation that transforms the set of reference samples into a new coordinate system such that the greatest variance, by any projection of the data, comes to lie on the first coordinate, the second greatest variance on the second coordinate, and so on. We apply the PCA on a set of patches containing only the targeted tool(s). We propose to use the M first principle components, having percentage of variance tends towards 99%, as filters to extract the features from each patch. In the coming sections, we refer to these filters as w_k.

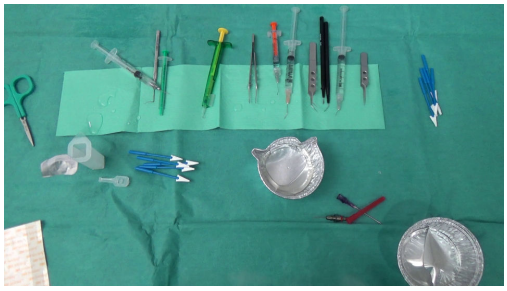
4.1.1.2 Change Descriptor

Considering the fact that the tools are being displaced on the tray, it is most likely possible that a patch P_i^f at position x_i^f in I^f will be found at position $x_i^f + d$ in I^l, as a patch P_j^l, where d is the displacement distance. A window (i.e. big patch) W^l centred on x_i^f of I^l is explored to find the patch P_j^l. P_j^l is defined as the patch whose feature vector V_j^l minimizes the Euclidean distance with V_i^f, the feature vector extracted from P_i^f. The *change* descriptor vector C_i of P_i^f implies looking for the most similar patch P_j^l in W^l in I^l and then describing it by the difference between feature vectors, as formulated in the equation:

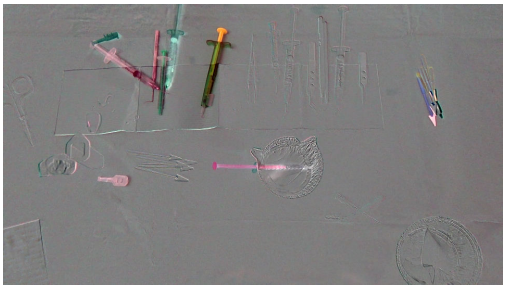
$$\begin{aligned} C_i = V_i^f - V_j^l \quad / \quad j = \arg\min_k \left(dist\left(V_i^f, V_k^l\right)\right) \\ i \in \left\{1, ..., n\right\} \; in \; I^f, \; k \; \in \left\{1, ..., m\right\} \; in \; W^l \end{aligned} \tag{4.1}$$



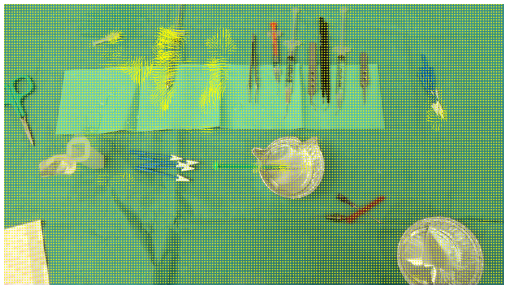
(a) Image right before a motion is detected



(b) Image right after a motion is stopped



(c) Image (b) - image (a)



(d) Optical flow result

(e) Optical flow registration

Figure 4.1: Summary of optical flow application. (a) and (b) represent the last image before a motion is detected and the first image after a motion is stopped, respectively. (c) The tools colored refer to the objects that have been put on and moved between the two images. Gray background indicates nothing moved. Edges of the objects show a sparse motion. (d) Yellow indicates the value and the direction of the Farnebäck optical flow calculated for each pixel [Farnebäck, 2003]. (e) Optical flow registration on image (c).

where $dist(V_i^f, V_k^l)$ is the Euclidean distance between the two feature vectors, n is the number of possible patches in I^f and m is the number of possible patches in the window W^l of I^l. Practically, this difference is large in case of tool appearance (no match is found in I^l), whereas it is close to zero in case of tool motion.

Without loss of generality, this section describes only the concept of the appearance of tools on the tray (tools put on the tray). To detect the appearance of tools in one patch from I^f, the corresponding patch in I^l is selected, then these patches are compared. To detect the disappearance of tools (tools taken from the tray), we simply swap the I^l and I^f images.

4.1.1.3 Classification

For each patch in I^f, we compute the *change* descriptor vector C, which implies looking for the most similar patch in I^l. The system is trained and tested using leave-one-out cross-validation. In other words, while processing the test image, all other images are used as training set. Here, we use a binary classification to detect the difference between a tool change and other changes. For each patch in the training dataset, the tool change probability is defined as the percentage of pixels inside the patch that belong to a tool put on or taken from the tray. Given a patch in the test set, the K nearest neighbours from the training set are searched for: the patch probability is defined as the average of the tool change probability among the nearest neighbours. In order to improve the computational time, we follow a coarse-to-fine configuration by starting with large patches and subdivide them if and only if the tool change probability is greater than 0% and so on until the desired patch size is reached.

4.1.1.4 Optimization

In this study, we introduce five parameters to be optimized. K indicates the number of the nearest neighbors to be taken into consideration. P_{min} is the smallest patch size in the list of patch sizes. τ is the scale factor used to go from a scale level to another. L is the number of the scale levels to be run and last but not least S is the window size of W^l. To find the optimal value for these discrete parameters, a discrete version of the *Particle Swarm Optimization* (PSO) algorithm was used here, called D-PSO. For a comprehensive review of D-PSO, we refer the reader to [Datta and Figueira, 2011].

4.1.2 Surgical Tray Actions Dataset

We use 36 out of 50 videos to ensure there are no noisy data produced by the plastic bag (see section 3.3.3). 36 surgeon actions were selected randomly, one per video. Two images were captured for each action, one right before it, the other one right after it. Those images were manually segmented by delineating the boundaries of the target tool put on or taken from the tray. The tools that were simply displaced were not segmented. Example of images that were manually segmented are given in Fig 4.3(c)(g).

	Malignant	Benign
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Table 4.1: Possible outcomes of a binary classifier benign/malignant.

4.1.3 Evaluation Metrics

The performances were evaluated in terms of area under the ROC curve (A_z). The ROC curve, also called the $sensitivity/specificity$ curve, makes it possible to evaluate the performance of a binary classifier. A binary classifier outcomes are labelled either as positive (P) or negative (N). There are four possible outcomes from a binary classifier. For instance, consider a pathology diagnosis test that seeks to determine whether a tumour is benign or malignant, the possible outcomes of a binary classifier are presented in Table 4.1. A true positive (TP) is when the outcome from a prediction is (P) and the actual value is also (P); however if the actual value is (N) then it is called a false positive (FP). Conversely, a true negative (TN) is when both the prediction outcome and the actual value are (N), and false negative (FN) is when the prediction outcome is (N) while the actual value is (P). As illustrated in Fig 4.2, the ROC curve represents the rate of TP ($sensitivity$) as a function of FP $\big(1 - specificity\big)$:

$$sensitivity = \frac{TP}{TP + FN} \tag{4.2}$$

$$1 - specificity = 1 - \frac{TN}{TN + FP} \tag{4.3}$$

The goal is to have a model that stands at the upper left corner of the curve, which is basically getting no false positives – a perfect classifier. In practice, the classifier should have a very good sensitivity (few FN cases), while being very specific (few FP cases). Therefore, the area under the ROC curve is generally between (random classifier) and (perfect classifier) and the objective is to have the largest possible area under the ROC curve.

4.1.4 Experimental Results

Two classification tests were conducted: one using the handcrafted features and one by applying the filters w_k on the patches to extract the learning features. Patch-level classification results of detecting the tools put on or taken from the tray are presented in Table 4.2 in terms of the area under the ROC curve (A_z). The probability of a tool change in the patches was used to compute the A_z. The A_z presented is the mean over the 36 test images. Observing the results with handcrafted and learning features, one can notice that the learning features outperform the handcrafted features. Such results are expected since color and edge information only consist of a mean of pixel values contained in the frame patches and the surgical tools resemble strongly, e.g. the knifes and the canulas. Nevertheless, these features still contain discriminative information taking into consideration the classification performance $\big(A_z = 0.947\big)$ obtained using these features. In contrast, the PCA-based features

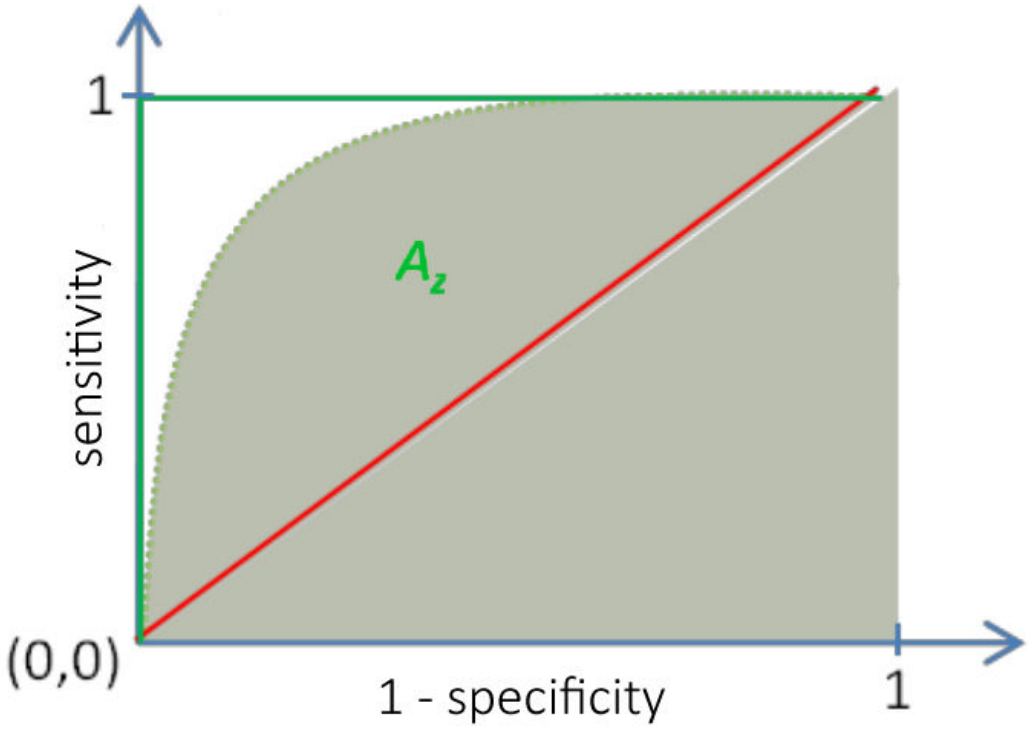


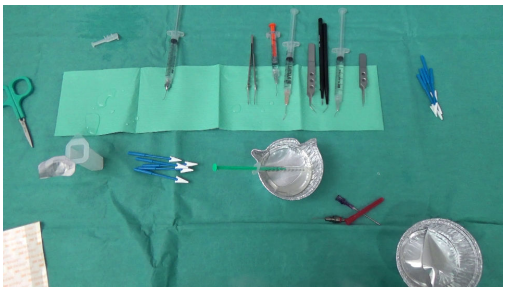
Figure 4.2: ROC curve presentation. Red line is the random representation , green lines represent perfect classifier and the dashed curve is an example of ROC curve with an area under it A_z.

yield higher results with $A_z = 0.959$**.** These features outperforms the handcrafted features due to the higher level of information contained in them, leading to better representation of the tools changes compared to the color and edge information.

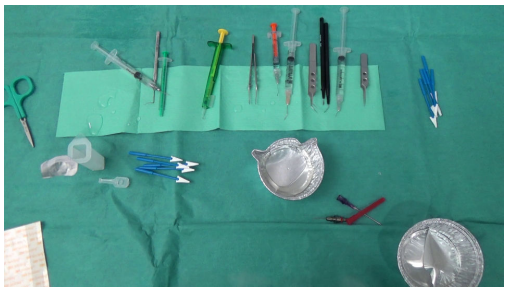
In Fig. 4.3, we show an example where the tools put on or taken from the surgical tray are perfectly detected, whereas the tools moved over the tray were not detected. But one limitation, presented in Fig. 4.3(h), where the tools put on or taken from the tray are detected as well as to the tools that were simply moved from one place to another. These tools are seen under a very different viewpoint in I^l and I^f. Indeed, this issue may due to the inherent properties of the features used in this study, i.e. they are not invariant to rotation and translation. Additionally, this limitation is due to the lack of annotated training data since the training set is only 35 images.

4.1.5 Change Detection Conclusion

Here, we have studied the feasibility of the proposed pipeline to detect the tools put on or taken from the surgical tray. We have also presented a feature change descriptor which learns conceptually the difference between the tools that were merely moved on the tray from the vanished/appeared tools. To evaluate the pipeline, we have extracted a subset of images from the surgical tray videos dataset and we applied leave-one-out cross validation approach. Despite the various visual chal-

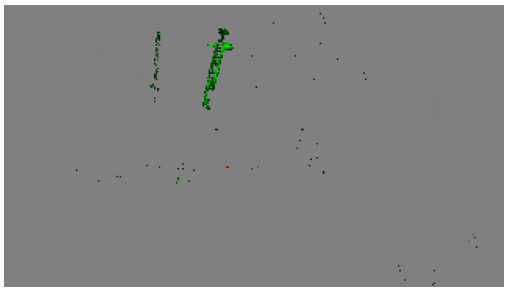


(a) Sample of I^l

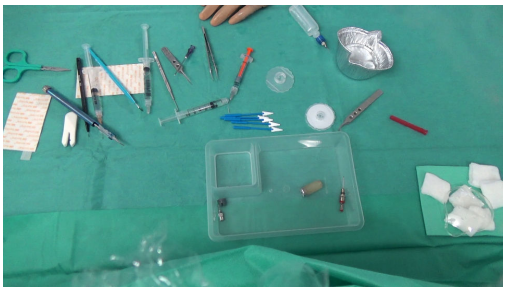


(b) Sample of I^f

(c) Manual segmentation of images (a) and (b)



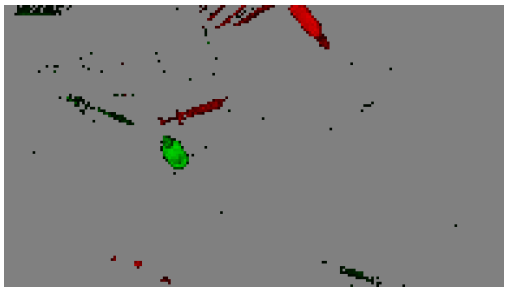
(d) Result of tool detection



(e) Sample of I^l

(f) Sample of I^f

(g) Manual segmentation of images (e) and (f)



(h) Result of tool detection

Figure 4.3: Two examples of tools detection: a success and a failure. (a), (b) and (c), (d) are two examples of surgical actions. In (d), (e), (h) and (i) gray indicates nothing moved, red level indicates a high probability of having a tool taken from the tray, green level represents high probability of having a tool put on the tray and black represents low probability of having a tool put on or taken from the tray.

Type	K	τ	S	P_{min}	L	P_{sizes}	A_z
Handcrafted features	89	4	81	5	3	[5;20;80]	0.947 $\pm$0.045
Learning features	99	2	90	8	3	[8;16;32]	0.959 $\pm$0.04

Table 4.2: Performance A_z of detecting the tools put on or taken from the tray using handcrafted and learning features.

lenges inherent in the surgical tray videos, our proposed pipeline shows promising performance for detecting the tools changes on the surgical tray. The experimental results shows the ability to carry out the task with high performance. However, the limitations of this method are strongly related to the features used and to the scant amount of data used to evaluate the pipeline. This limited amount of data is due to the complexity of annotating the changes over all the surgical tray videos.

Ideally, recognizing the tools changes is practically more relevant to our aim, however, telling only that a tool (whatever it is) is put on or taken from the tray can be deemed a significant information as well. The dearth of annotated data hinders the recognition of the tools changes task and makes it substantially impractical due to the complexity of the task on such dataset.

4.2 Tool Presence Detection

Detecting tool presence, at each instant of the surgery, is the second strategy applied in this thesis in order to tell the tools that are probably present in the surgical field. The task of tool presence detection is to provide a binary information denoting the presence of tools of interest. A thorough search of the relevant literature reveals that the previous studies tackling the surgical tool presence detection over the surgical tray using only visual information are scarce with mediocre results. In fact, the tool presence detection task is not a trivial task over the surgical tray. The visual challenges inherent in the surgical tray videos, described in section 3.3.3 (e.g. low distinctive patterns of tools), results in a low inter-class variability which is challenging for any classification problem. Here, we study the feasibility of performing the tool presence detection task on the surgical tray videos.

In the following, we describe the pipeline used to perform the surgical tool presence detection task. Then, we describe the experimental datasets used to evaluate the proposed method. After presenting the experimental results, we present the conclusions of performing the proposed pipeline to detect the tool presence.

4.2.1 Methodology

A patch-based approach, similar to the *change* detection one, is applied to detect the tool presence. The pipeline of this method consists of: (1) features extraction step to represent each patch of the image by a feature vector. (2) classify each patch into the tool it belongs to. The first step is identical to the feature extraction step of the *change* detection method, described in section 4.1.1.1, however, in this strategy, we only focus on the learning features to evaluate this method. In addition, it is

necessary to point out that using this method we can recognize the tools present on the tray at each instant of the surgery and, consequently, the tools put on or taken from the surgical tray.

4.2.1.1 Classification

For each patch in an image, we first compute the learning feature vector. In this study, we use a one-versus-all multi-label technique to detect the tool presence because each image contains many tools at the same time. The system is trained using leave-one-out cross-validation to only optimize the parameters. For each patch in the training set, the tool presence probability is defined as the percentage of pixels that belongs to the targeted tool. Local-based optimization is used during the training stage. At inference time, the classification is based on k-NN regression. Given a patch in the test set, we use the optimal values of the parameters, found during the training, to find the K nearest neighbours from the training set: the patch probability is defined as the average of the tool presence probability among the nearest neighbours. Similar to *change* detection, a coarse-to-fine configuration is followed for computation purposes.

4.2.1.2 Optimization

We optimize the same parameters as the ones used in the change detection approach K, P_{min}, τ and L (detailed in section 4.1.1.4) with the exception of S which is dedicated for the block-matching approach. Additionally, D-PSO is used to optimize these unknowns during the training stage.

4.2.2 Experimental Datasets

The dataset was divided into a training set (25 videos) and a test set (25 videos). The division was done at random with the exceptions of: 1) each tool appears in the same number of videos from both subsets (plus or minus one) and 2) the test set only contains videos from surgeries performed by the renowned expert. In this study, 35 images were extracted from the training set containing all the targeted tools. The tools are present approximately in all the training images. They were segmented manually by filling the boundaries of each tool by a specific color.

4.2.3 Evaluation Metrics

The evaluation metric used for this task is the area under the ROC curve A_z. It is identical to the metric used to measure the system's performance for *change* detection, as presented in section 4.1.3.

4.2.4 Experimental Results

To streamline the process, the experiments were done on the test set at 1 fps and for only the most common tools in the cataract surgery. In Table 4.3, we show the results in terms of A_z for this subset of tools including the optimal values of

the parameters. For a graphical representation of the results, an illustration of the A_z obtained is shown in Fig.4.4. The sum of the probabilities computed for the image patches is the criteria used to gauge the A_z on the test set. The average performance of the system is m$A_z = 0.6$. One can obviously notice that the tool presence detection results are not good for ten tools out of eleven. The exception is the cotton tool with $A_z = 0.961$. Also, the patch sizes P_{sizes} are correlated with the smallness and thinness of the targeted tools. Interestingly, the cotton has the lowest value of the nearest neighbours K. This verifies the significance of the filters obtained for this tool, resulting in high A_z. In Fig.4.5, two examples of cotton detection are presented. The first row represents a success where the model is able to perfectly detect the cottons without any outliers. However, the second row shows decent results where other objects are detected as well as to the cottons. For

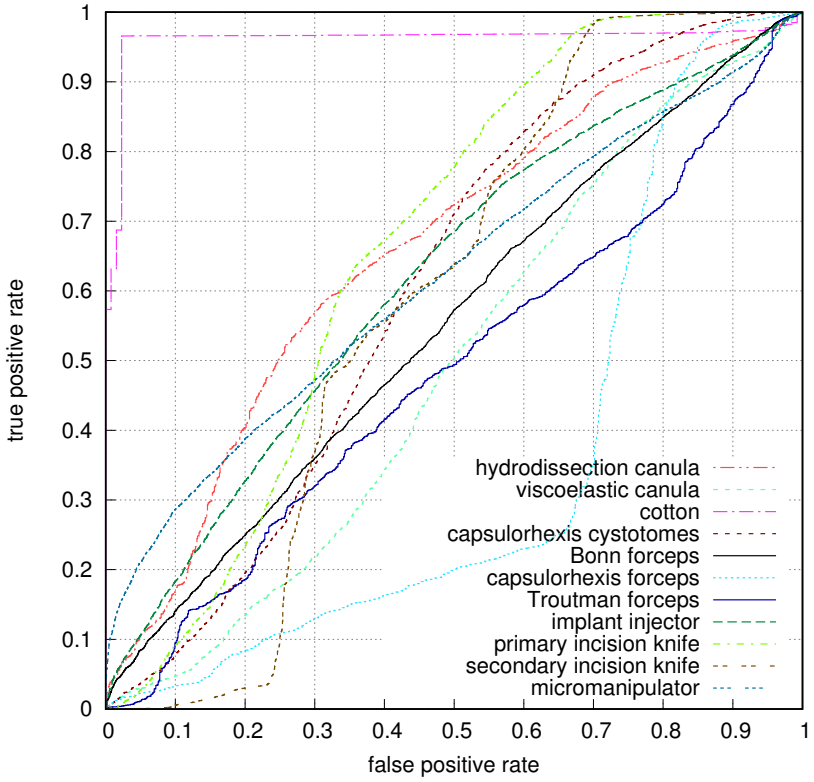


Figure 4.4: ROC curves for the most frequent tools used in the cataract surgery.

the other tools, the models are clueless, as shown in Fig.4.6 and 4.7. The classifiers detect approximately all the tools at once without being able to separate them. Due to the similarities between the targeted tools, the visual learning features extracted are not highly discriminative to differentiate a targeted tool from anything else over the tray. For instance in Fig.4.5(d) and Fig.4.7(h), the learning features extracted seems to be highly correlated to the color features rather than any other sort of information.

Moreover, it is necessary to note that these results are obtained through frame-wise classification that only rely on a small number of examples of the tools which have high inter-class similarities. This limitation is due to the complexity of segmenting

all the tools in such large dataset. This constraint adversely affects the results obtained and it can be considered the major downside to the effectiveness of the proposed method.

Tool	K	τ	P_{min}	L	P_{sizes}	A_z
hydrodissection canula	91	3	2	2	[2;6]	0.659
viscoelastic canula	100	3	7	1	[7]	0.491
cotton	18	3	2	2	[2;6]	**0.961**
capsulorhexis cystotome	62	3	6	1	[6]	0.606
Bonn forceps	100	5	4	1	[4]	0.552
capsulorhexis forceps	85	5	3	1	[3]	*0.354*
Troutman forceps	83	3	2	2	[2;6]	0.485
implant injector	88	4	9	1	[9]	0.619
primary incision knife	100	2	5	2	[5;10]	0.663
secondary incision knife	56	2	2	2	[2;4]	0.586
micromanipulator	100	3	10	1	[10]	0.618
Average (mA_z)						0.6

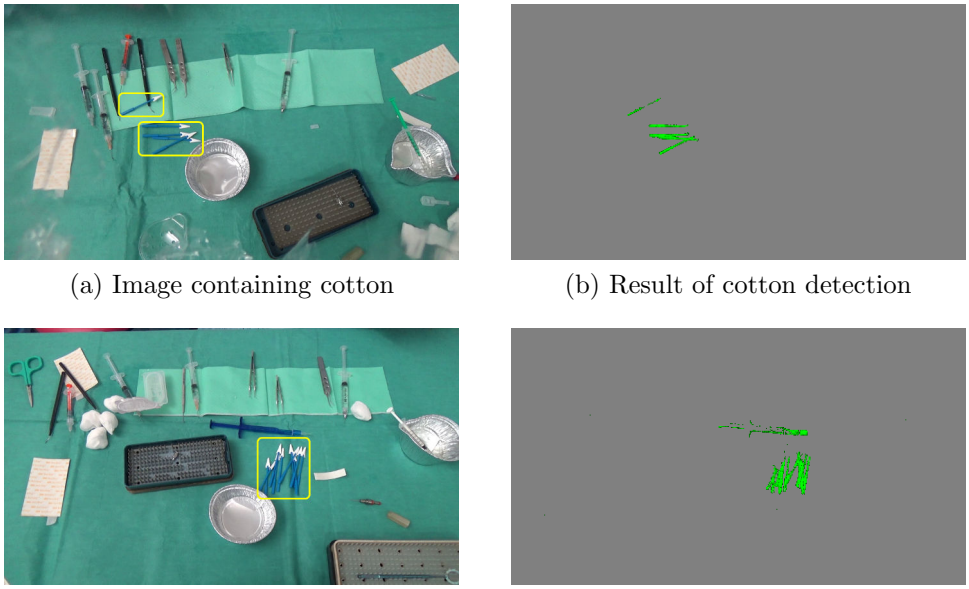
Table 4.3: Performance A_z of surgical tool presence detection using learning features. The best object detected is presented in bold and the least one is presented in italic.

4.2.5 Tool Presence Detection Conclusion

Here, we have addressed the surgical tool presence detection task in the surgical tray videos. Into the proposed pipeline, we used the visual learning features described in section 4.1.1.1. The training is done on a patch-level of the small set of images extracted from the 25 training videos. To evaluate the method, we perform the surgical tool presence detection on image-level for the 25 testing videos at 1 fps. The evaluation is performed on a subset of tools which are mostly used in a normal conduct of the cataract surgeries. The experimental results show very good results for only one tool (cotton with $A_z = 0.96$). By reason of its unique shape and color, the cotton has been perfectly represented in the learning features extracted, despite the modest size of the training dataset. However, for all other tools, the models performed poorly at inference time because of the non-discriminative features used to perform the classification. This might due to the high similarities between the tools and the scant amount of training data.

4.3 Summary

In this chapter, we have presented a patch-based template matching technique to handle the surgical tool detection over the surgical tray. It has shown incapacity in solving such problem. The solution was applied on two different strategies to obtain the tool information signals. The first strategy is the change detection where we detect only the changes occurring along the surgery. The second strategy is to detect



(a) Image containing cotton

(b) Result of cotton detection

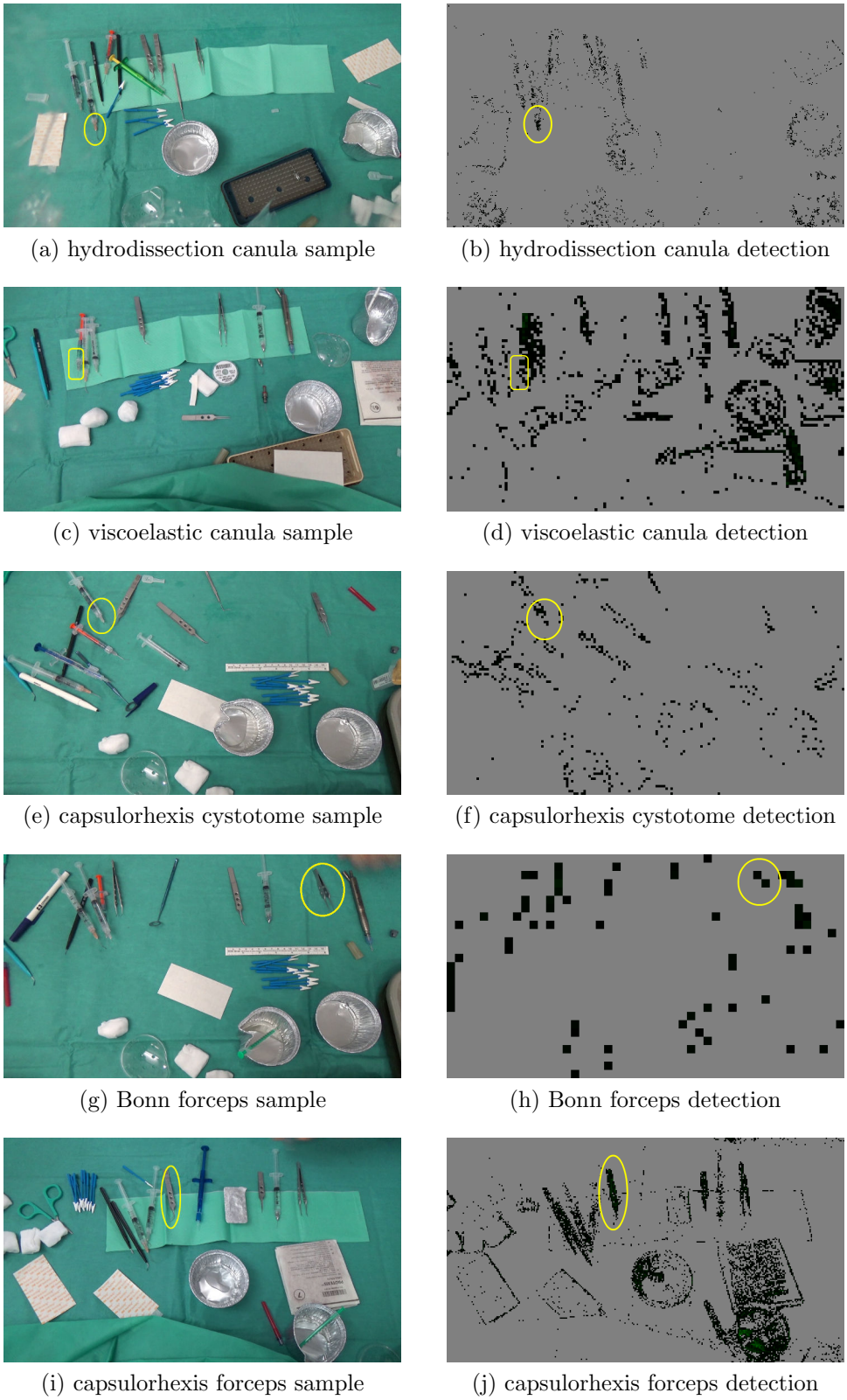
(c) Image containing cotton

(d) Result of cotton detection

Figure 4.5: Two examples for cotton detection: a success and a failure. In (a) and (c), tools bounded box in yellow are the targeted tools in each image. In (b) and (d), pixel value represents the probability of having the targeted tool in the patch. Gray indicates probability equal zero and green indicates high probability.

the tools presence over the tray at each instant of the surgery. The solutions proposed for both strategies resemble strongly. In regards to the change detection, the solution showed good performance in detecting that a tool has put on or taken from the tray. However, the complexity of obtaining a ground truth for all tools changes in such huge dataset impedes the recognition of these changes, i.e. tell which tools have been put on or taken from the surgical tray. For detecting tools presence, the solution has deficiently performed the task. The results are likely being impacted by two reasons: (1) the PCA-based features were not decently discriminative. (2) the lack of annotated data for the training stage adversely affected the results. In addition, the solution was not computationally efficient to run the analysis over a large dataset such as the cataract surgery dataset, e.g. one frame needs at least tens of seconds to be processed. Therefore, these roadblocks impede a practical solution using this patch-based approach. In addition, the *homography*-based solution, presented in Appendix A, has performed yet poorly in simple tool detection scenarios.

Apparently, the template matching-based solutions have significant limitations on the tray videos. In this regard, a different kind of solution is required to have more appropriate representation of the tools. At this moment of the thesis, template matching techniques were not any more the best-suited approach for pattern recognition after the emergence of deep learning. Using deep learning, no need any more for handcrafted or PCA-based (i.e. shallow learning based) visual features because it automatically learns deep features from the data. Additionally, using deep



(a) hydrodissection canula sample

(b) hydrodissection canula detection

(c) viscoelastic canula sample

(d) viscoelastic canula detection

(e) capsulorhexis cystotome sample

(f) capsulorhexis cystotome detection

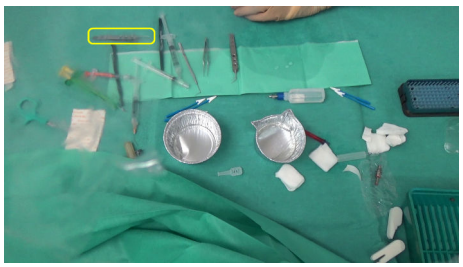
(g) Bonn forceps sample

(h) Bonn forceps detection

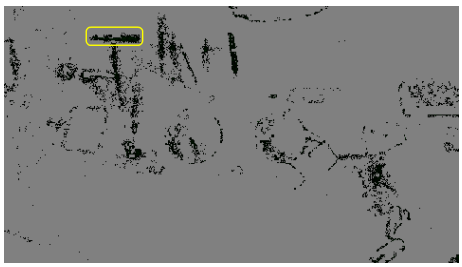
(i) capsulorhexis forceps sample

(j) capsulorhexis forceps detection

Figure 4.6: Examples of surgical tools detection. Tools bounded box in yellow are the targeted tools in each image. **Right:** pixel value represents the probability of having the targeted tool in the patch. Gray indicates probability equal zero, green indicates high probability and black indicates low probability.



(a) Troutman forceps sample

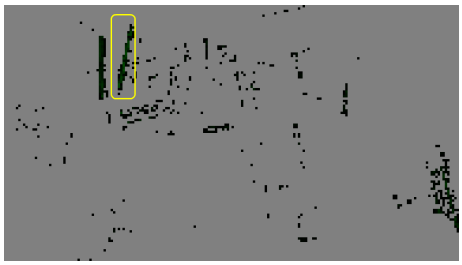


(b) Troutman forceps detection

(c) implant injector sample

(d) implant injector detection

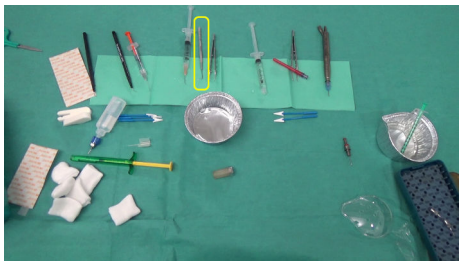
(e) primary incision knife sample



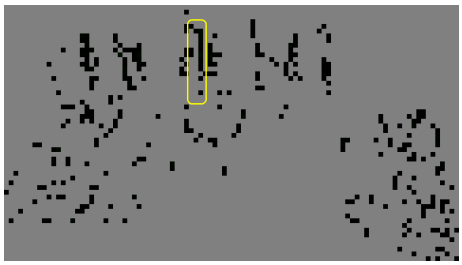
(f) primary incision knife detection

(g) secondary incision knife sample

(h) secondary incision knife detection



(i) micromanipulator sample



(j) micromanipulator detection

Figure 4.7: Figure 4.6 (Cont.).

learning, it is possible to perform a weakly-supervised approach on image-level and not on pixel-level, which makes the ground truth data acquisition easier, leading to a more data fed into the training stage and to probably better performance. This allows us to perform the surgical tool detection on the tool-tissue interaction videos and the surgical tray videos while being computationally efficient. The template matching solution, proposed in this chapter, is the first method that tackles tool detection on the surgical tray videos in the literature. Regardless of its efficiency, in this thesis, it is considered as a reference solution to be compared later with the deep learning-based solution.

"It always seems impossible until it's done."

Nelson Mandela

5

Surgical Tool Detection in Surgical Videos using Deep Learning

Chapter Content

Swiftly, deep learning has gained a lot of attention and has absolutely dominated computer vision over the last few years. Subsequently, surgical tool detection in the cataract surgery videos was geared towards a deep neural network-based solution. In this chapter, we present a surgical tool detection pipeline based on deep learning models for both video types: tool-tissue interaction and surgical tray videos. For the surgical tray, we present a pipeline for the two strategies followed in the previous chapter: (1) detecting only the tools changes along the surgery: tools put on or taken from the tray and (2) detecting tool presence at each instant of the surgery. In regards to the tool-tissue interaction videos, a similar approach to the surgical tray tool presence strategy is presented in this chapter. In addition to the dataset of real-world cataract surgeries (described in chapter 3), referred in this thesis as RW dataset, we propose to use simulated surgical tray datasets, which are generated manually to alleviate the inherent challenges of the RW surgical tray dataset.

This chapter is structured in five sections. First, we present a comprehensive review on neural networks. In section 5.2, we describe the most common network architectures used for processing images. The section 5.3.4 discusses the change detection on the surgical tray using deep learning based method. The tool presence detection pipelines for both video types are presented in section 5.4. Before summarizing this chapter, we perform the surgical tool presence detection on simulated surgical tray datasets, described in section 5.5.1.

5.1 Deep Neural Networks

Surgical tool detection can be formulated as requiring a computer to perform a mapping $f : X \rightarrow Y$ where X is an input space and Y is an output space. In this thesis, X is the space of images (extracted from videos) and Y can be an interval of $[0, 1]$ representing the probability of a tool appearing in the image. To specify the function f, a data-driven approach with a supervised learning paradigm is the best-suited to address the surgical tool detection problem. In this section, the training examples are denoted by $\{(x_1, y_1), ...(x_n, y_n)\}$ where $(x_i, y_i) \in X \times Y$.

This section provides the necessary technical background on neural networks. For a more thorough introduction, we recommend the Deep Learning Book [Goodfellow et al., 2016]. Primarily, we describe the vanilla neural networks in section 5.1.1. In section 5.1.1.1, we discuss few optimization strategies in order to successfully train the neural network models. In section 5.1.2, we present a successful type of deep neural networks for handling data with some spatial topology (images, videos, 3D

voxel data and character sequences in text), called convolutional neural networks (CNNs). We end this section by discussing transfer learning, which is nowadays one of the most common methodologies used in medical image analysis.

5.1.1 Vanilla Neural Network

The vanilla neural network has originally been inspired from the biological neural systems, where neurons are interconnected and pass messages from one to another. The neurons are the computational units of the human brain. Intuitively, each neuron receives a list of input signals and produces an output signal after performing some computations based on the strength of each input signal. The input strength are learnable and control the impact of one neuron on another. This is the description of a basic biological neural model. Similarly, the artificial neural networks (ANNs) are constructed. For each neuron, the input signals are either the input data (images, text etc.) or the output of other neurons. The strength of each input signal is the learnable weights, referred as w. The firing rate of a neuron is represented by an activation function, referred as a applied on the input signals to produce an output. A matrix representation of the computation inside the neuron can be formulated as:

$$z = w \cdot \mathrm{x} + b \tag{5.1}$$

where x are the input signals, w and b are the weights and the bias of the neuron, respectively. In other words, each neuron performs a dot product with the input and its weights, adds the bias and applies the activation function representing the non-linearity. An illustration of an artificial neuron is shown in Fig.5.1(a).

In the realm of artificial neural networks, the neurons are connected in an acyclic graph, as illustrated in Fig. 5.1(b). The neural network models are often organized into distinct layers of neurons, which are not connected to one another in each layer. In the literature, more than two-layer neural networks is typically considered a deep neural networks, thus the name of *deep learning*. A network is organized in a layer-wise manner, which is mathematically represented by composing together many different functions. For instance, in Fig. 5.1b, we can have three different functions, $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$, representing the three different layers, connected in a chain, to form:

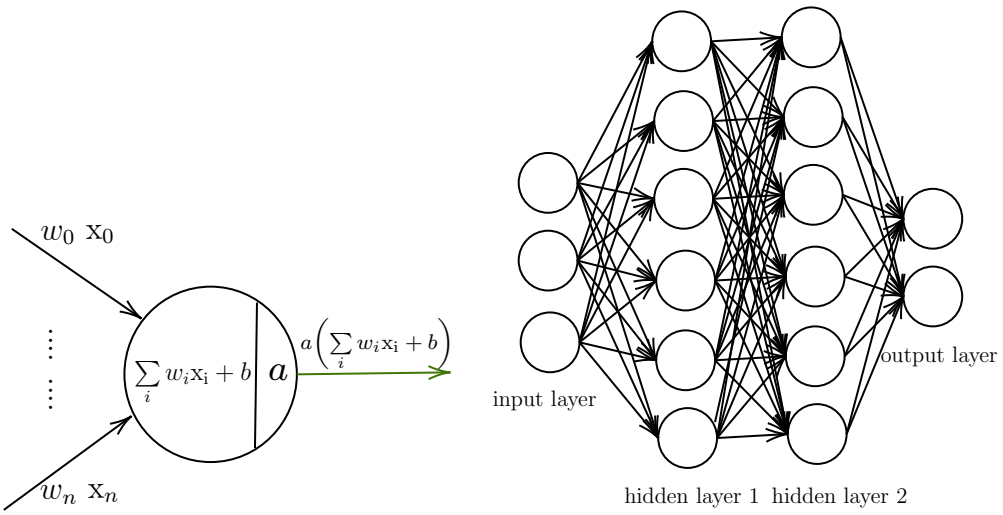
$$f(x) = f^{(3)}\Big(f^{(2)}\Big(f^{(1)}(x)\Big)\Big) \tag{5.2}$$

where $f^{(1)}$ is the function of the first layer of the network, $f^{(2)}$ is the function of the second layer of the network and so on.

For a network with L layers, the function of the l-th layer consisting of N_l neurons, denoted as $f^{(l)}$, is expressed as:

$$f^{(l)}(x) = W_l \cdot a\Big(f^{(l-1)}(x)\Big) + B_l \tag{5.3}$$

where $W_l = [w_l^1, ..., w_l^{N_l}]^T$ and $B_l = [b_l^1, ..., b_l^{N_l}]^T$ are the weight matrix and the bias vector of the l-th layer, w_l^m and b_l^m are the weight and the bias of the m-th neuron of the l-th layer. $f^{(0)}(x)$ is the input data of the neural network. $a(.)$ is the activation function. This process is called the forward pass. The activation function



(a) Illustration of biological inspiration behind the single artificial neuron.

(b) An example of 3-layer neural networks.

Figure 5.1: Representation of one single neuron and a complete neural network. Neurons in one layer have connections to all neurons in the next layer with the exception of the output layer. To evaluate activations of all neurons in a single layer, a matrix multiplication is a relevant representation thanks to this way of arrangement of neurons.

a is typically chosen to be a function that is applied element-wise. Various kinds of non-linear activations functions were used in artificial neural networks:

- **Hyperbolic tangent:** $a(t) = \dfrac{e^t - e^{-t}}{e^t + e^{-t}}$
- **Logistic sigmoid:** $a(t) = \sigma(t) = \dfrac{1}{1 + e^{-t}}$
- **Rectified linear unit (ReLU):** $a(t) = max(0, t)$. In modern neural networks, the default recommendation is to use the ReLU as activation function, as stated in [Goodfellow et al., 2016].

In ANNs, the values of the output layer, denoted as $\hat{y} = [\hat{y}^1, ..., \hat{y}^K]$ with K is the number of neurons in the output layer, are commonly referred as logits. In any classification problem, three different kinds of classifications are possible: binary, multi-class and multi-label classification. For binary classification, the output layer consists of one node ($K = 1$) representing the confidence of an input data x belonging to the corresponding class. To evaluate the output of the network in this case, a loss (cost) function is required. It is scalar-valued loss function $L(\hat{y}, y)$ that measures the disagreement between a predicted label $\hat{y}_i = f(x_i)$ and a true label y_i. It is usually the objective function to optimize. The most commonly used loss function in the classification settings is the cross-entropy loss, which has the following equation in

any binary classification problem:

$$L(\hat{y}, y) = -\frac{1}{n}\sum_{i=1}^{n}\Big[y_i \ln\big(\sigma(\hat{y}_i^1)\big) + (1-y_i)\ln\big(1-\sigma(\hat{y}_i^1)\big)\Big] \tag{5.4}$$

where n is the number of training examples, $\sigma(.)$ is the sigmoid function and $\hat{y}_i^1$ is the predicted value of an input data x_i for the only class in the classification.

For multi-class classification problem where the output layer amounts to K nodes. The cross-entropy loss function is then expressed as:

$$L_{W,B}(\hat{y}, y) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\Big[y_i^k \ln\big(\sigma(\hat{y}_i^k)\big)\Big] \tag{5.5}$$

where y_i^k is the ground truth of input data x_i for the class k and $\hat{y}_i^k$ is the predicted value of x_i for the class k.

In regards to the multi-label classification, binary relevance is the standard one-vs-all scheme applied to multi-label classification. For instance, for J labels, the neural network performs independently the training of J binary classifiers. The cross-entropy loss function can be expressed as follows:

$$L_{W,B}(\hat{y}, y) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J}\Big[y_i^j \ln\big(\sigma(\hat{y}_i^j)\big) + (1-y_i^j)\ln\big(1-\sigma(\hat{y}_i^j)\big)\Big] \tag{5.6}$$

where $y_i = [y_i^1, ..., y_i^J]$ is the ground truth of an input data x_i.

5.1.1.1 Optimization

The optimization of a neural network corresponds to searching over a set of candidate functions F and finding the most consistent one f^* with the training examples. The objective is then to approximate some function $f^* \in F$, that minimizes the expected loss over the training data, to map elements of X to Y:

$$f^* \approx \arg\min_{f\in F}\frac{1}{n}\sum_{i=1}^{n} L\Big(f(x_i), y_i\Big) \tag{5.7}$$

In other words, it amounts to finding W and B that provide us the lowest loss, where $W = [W_1, ..., W_l]$ and $B_l = [B_1, ..., B_l]$ for a neural network of L layers. During the training of a neural network, $f(x)$ is driven to match $f^*(x)$. The training process provides approximate examples of $f^*(x)$, evaluated at different training points. The prediction of each input data x is a label $y \approx f^*(x)$. In practice, the loss functions can be minimized using gradient descent methods. To figure out which direction to alter the parameters, the rate of change of the loss function with respect to the weights is required, subsequently computing the derivative of the loss function. The gradient descent consists of: (1) evaluating the gradient using backpropagation. (2) updating the networks parameters W and B during the training by taking a small step in the direction of the negative gradient. The parameters updates can be formulated as:

$$W^{'} = W - \nu \cdot \nabla_W L(\hat{y}, y) \tag{5.8}$$

$$B^{'} = B - \nu \cdot \nabla_B L(\hat{y}, y) \tag{5.9}$$

where ν is the step size hyperparameter, known as "learning rate".

5.1.1.2 Regularization

There might be plenty of f^* that satisfies Equation (5.7). In other words, there may be different functions f^* that all achieve the lowest possible loss, however, their generalization outside the training data could vary. This poses a challenge for the optimization process, called overfitting, where a function performs very well on the training data but does not generalize on the inference data. In this regard, a regularization term is added to the optimization process that encodes preference for some functions over others, regardless of their fit to the training data. The optimization Equation (5.7) with the addition of regularization is expressed as:

$$f^* \approx \arg\min_{f \in F} \frac{1}{n} \sum_{i=1}^{n} L\Big(f(x_i), y_i\Big) + R(W) \tag{5.10}$$

where R is a scalar-valued function. This function shrinks the weights towards zero. In fact, it discourages the learning of complex models, by penalizing them, to sidestep the overfitting. In other words, the regularization technique can be seen as a penalty function to quantify complexity of the model. The more complex the models are the greater the penalty will be. The regularization can be marginally justified by applying the principle of Occam's razor in the optimization process, which is stated as: *"Suppose there exist two explanations for an occurrence. In this case the simpler one is usually better".* In this context, the most common technique used nowadays is L2 regularization, which can be expressed as follows:

$$R(W) = \lambda \sum_{l=1}^{L} \|W_l\|_2^2 \tag{5.11}$$

where λ is a hyperparameter called "weight decay". It is a scalar value that reflects how much power is given to the regularization term to impact the optimization process.

5.1.2 Convolutional Neural Network

The typical ANNs have some limitations regarding the size of the input data fed into it and the type of this data. For instance, by stretching out the pixels of an image as an input vector of the ANN, we lose utterly the spatial information presented in the image. Intuitively, using a small input image containing some spatial information is the solution for such problem, which is the bottom line behind the convolutional neural networks (CNNs) [Lecun et al., 1998]. The CNNs are a special kind of neural networks for processing data that have spatial topology. They are currently the quintessential deep learning models. The CNNs are simply neural networks that use the mathematical operation convolution in a least one of their layers. They automatically detect the prominent features and classify them without any human supervision while being computationally efficient. The optimization of a CNN is similar to the ANN (described in section 5.1.1.1), in which the process is done by using mini-batches of training data. A mini-batch contains a subset of the training data. In addition, it is common to use the term "batch size" to refer

to the size of a mini-batch. Various optimization algorithms are currently used by the community [Kingma and Ba, 2014], i.e. Adam, RMSProp, Stochastic Gradient Descent (SDG), just to name a few. As illustrated in Fig.5.2, a typical CNN can have three types of layers: (1) Convolution layer. (2) Pooling layer. (3) Fully-connected (FC) Layer.

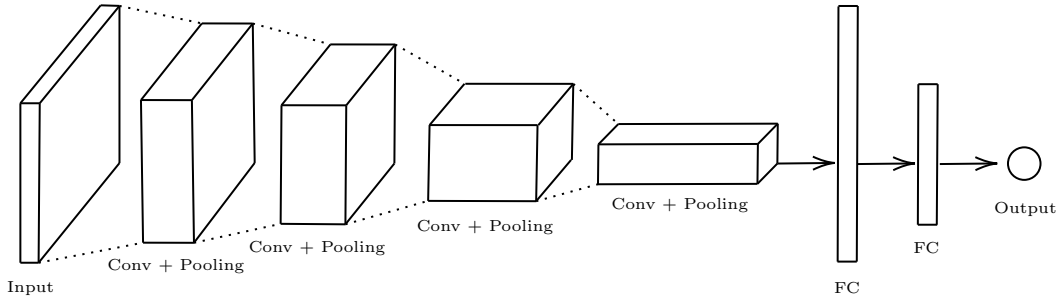


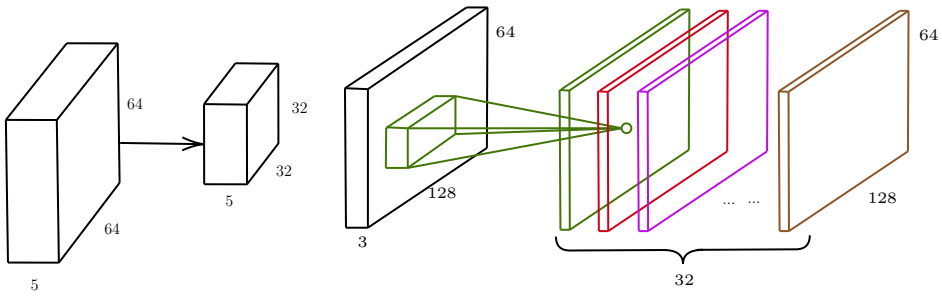
Figure 5.2: An example of a simple convolution neural network.

5.1.2.1 Convolution Layer

In order to take advantage of the spatial information, the convolution layer operates over volumes. All the convolution layers take volumes as input and produce volumes of feature maps (or receptive fields). The input shape of a convolution layer is $height(H) \times width(W) \times depth(D)$. The parameters of this layer consist of a set of learnable filters. Each filter produces a separate 2 dimensional feature map. The layer performs the same neuron computation as in Equation (5.1), with one main difference: each neuron in a convolution layer is only connected to a local volume of the input one, as illustrated in 5.3(b). In this example, an input volume of shape $64 \times 128 \times 3$ is converted to an output volume of shape $64 \times 128 \times 32$. Each output neuron is the result of the dot products (convolution operation) between the entries of the filter and a sliding window across W and H of the input volume. Every filter is small spatially (along W and H), but extends through the depth of the input volume. Interestingly, all neurons of one feature map share the same parameters (same filter), resulting in a reduction of the number of unknowns and ensuring translation invariance. Since convolution is a linear operation, the result of the convolution operation is passed through an activation function to add the non-linearity, i.e. ReLU. Therefore, the final values in the feature maps are the results of applying ReLU on the results of the convolution operations.

5.1.2.2 Pooling Layer

The pooling layers usually reduce the dimensionality, leading to a shortening of the training time. They have no parameters, thus the reduction of the number of the parameters to be optimized by the network. In fact, they downsample each feature maps independently while keeping the depth of the input intact. An example of a pooling layer is shown in Fig.5.3(a). In this example, a sliding window with stride of 2 pixels is used to pass over the input $64 \times 64 \times 5$ and simply takes one value to



(a) Pooling layer illustration

(b) Convolution layer illustration

Figure 5.3: Two types of layers in CNNs.

represent the window in the output $32 \times 32 \times 5$. The most common type of a pooling layer is the *max pooling* where the maximum value in the window is retained. It is also common to apply a pooling layer after a convolution layer in an alternating manner.

5.1.2.3 Fully-Connected Layer

The fully connected layer is a typical ANN. It is called fully connected (FC) since all neurons in a hidden layer are connected to all neurons in the next and previous layers. The fully-connected layer perform the classification task in a CNN. Using CNNs, the different types of classification strategies are addressed using the same Equations (5.4), (5.5) and (5.6) described in section 5.1.1.

5.1.3 Transfer Learning

Transfer learning is a design methodology within machine learning [Pan and Yang, 2010]. The concept is to use the knowledge learnt from tasks for which labelled data is available in other tasks where scant amount of labelled data is available. The employment of transfer learning is an attempt to start the generalization process of a model from patterns that have been learnt for a different task instead of starting the process from scratch, and probably leading to a faster convergence of the targeted model. In practice, training a sophisticated CNN from scratch on a complex dataset is relatively not common due to the complication of the task and to the limited resources available. Instead, it is possible to finetune a CNN, pretrained on a very large dataset, on the targeted dataset. Finetuning is a way of using the learnt parameters as an initialization for the targeted task. In fact, the earlier features of a CNN contain generic features (e.g. edge information) which can be advantageous to other tasks, but the more the layer is deeper, the features become progressively more specific to the classes of the original dataset. The finetuning can be applied by modifying the dense layers (FC layers) in a manner that the network output suits the targeted task and train only the classifiers. This approach is used when there is scant amount of data. A second approach is to include some or all the convolution layers with the dense layers to be finetuned during the training.

5.2 Network Architectures

Since the emergence of deep learning models, the ImageNet dataset [Deng et al., 2009] has become a well known dataset in the computer vision domain. Since 2010, this dataset was used in the context of the challenge ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It is acknowledged as being an event for benchmarking deep learning algorithm for object recognition. The task in the object recognition competition is to classify the images into one class or five classes of a thousand classes. In Fig.5.4, a complexity comparison of most successful deep learning networks on ImageNet is presented.

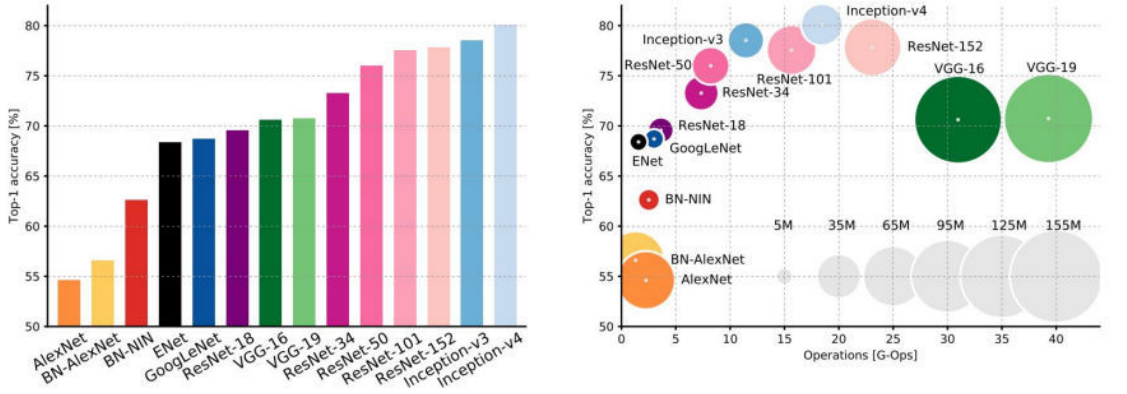


Figure 5.4: Courtesy of [Canziani et al., 2016]. Complexity comparison between top scoring deep learning networks for ImageNet classification task until early 2017. **Left:** top-1 validation accuracies for single-model architectures. **Right:** top-1 accuracy in function of the amount of operations (G-Ops: giga operations per second) required for a forward pass. The blobs size is proportional to the number of network parameters. The legend, reported in the bottom right corner, is spanning from 5×10^6 to 155×10^6 parameters.

However, research in deep learning proceeds rapidly so that a new best network architecture for ImageNet classification challenge is announced every few weeks to months, making it impractical to choose the best deep learning classification architecture. Therefore, in this thesis, we work with a list of the best network architectures proposed to date for image classification problems. These network architectures are described in details in the following sections.

5.2.1 Earlier Networks

In ImageNet ILSVRC 2012, AlexNet was the first CNN that won the classification competition. It achieved a large accuracy margin compared with the non deep neural networks methods. It consists of 5 convolutional layers and 3 FC layers. At that time, stacking more and more layers in a CNN seemed the best approach to get better performance. Driven by the significance of depth, VGG-19 [Simonyan and Zisserman, 2014b], a network of 19 layers, was one of the most performant network in ILSVRC 2014. Nevertheless, keep increasing the depth adversely affects the

convergence of the network because the gradient values become smaller and smaller as they are propagated from deep to shallow layers 1994. The vanishing/exploding gradient problem hinders the convergence of the model from the beginning. This problem has been predominantly alleviated by normalized initialization [Saxe et al., 2013] [He et al., 2015] [Glorot and Bengio, 2010]. However, another problem arises when the accuracy gets saturated and then degrades swiftly. This issue is unexpectedly not caused by overfitting and adding repeatedly layers leads to higher training error, as reported in [Srivastava et al., 2015] [He and Sun, 2015].

5.2.2 Residual Network

A solution for the degradation problem was proposed in Microsoft's research lab in Beijing [He et al., 2016b]. They introduced the concept of residual blocks, which have brought advantages with minor changes to the CNN architecture. In Fig.5.5, we show the difference between the normal transition between layers and a transition based on residual blocks. Formally, rather than optimizing $h(x)$ that maps few

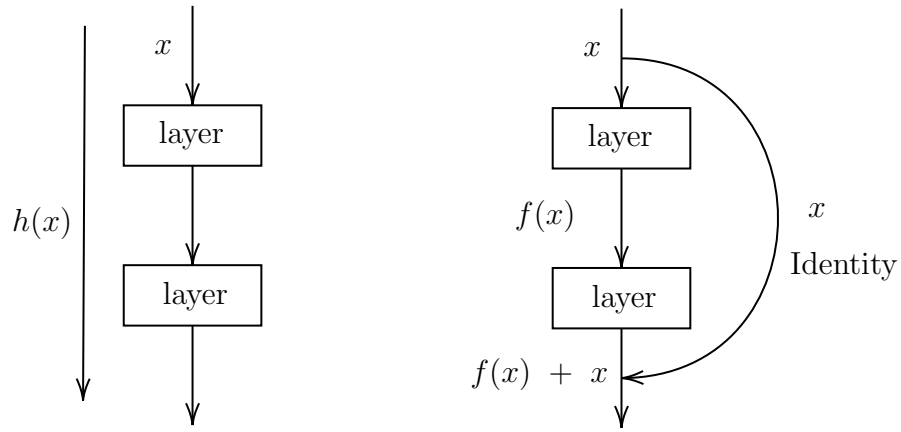


Figure 5.5: **Left:** Normal CNN. **Right:** Residual Linked CNN.

stacked layers, one can optimize an approximation of a residual function:

$$f(x) = h(x) - x \tag{5.12}$$

This shortcut connection $f(x) + x$ introduces neither extra unknowns nor computation complexity. Intriguingly, the optimization becomes easier since it provides faster convergence at the early training stage, as proven in [He et al., 2016b]. A residual network is then a series of 3×3 blocks of convolution layers followed by a pooling layer, in which a shortcut connection is added to link those blocks. The network ends with a global average pooling followed by a fully-connected layer. There are various versions of this network (ResNet-34, ResNet-50, ResNet-101 etc.) which differ only in the number of 3×3 convolution layers. However, the most common one is ResNet-152, which has won the 1^{st} place in the ImageNet ILSVRC 2015 classification competition. A generalisation of residual networks have earned as well the 1^{st} prize on ImageNet detection, ImageNet localization and Common Objects in Context challenge (COCO) 2015 competitions.

5.2.3 Inception Network

If the essence of residual networks is to go deeper, the inception networks are built to go wider. The inception networks were inspired from the work of Network in Network (NIN) [Lin et al., 2013] while taking into account the depth of the network. In addition to the network depth, the network width (i.e. the number of units at each layer) is also considered significant for improving the performance of the network. In [Szegedy et al., 2015b], they introduced a new level of layers organization in the form of "Inception module". To explore better the spatial information, the module acts as multiple convolution filters (i.e. 1×1, 3×3, 5×5 and 7×7 convolution filters), applied to the same input and their results are concatenated with the result of a pooling layer, as illustrated in Fig.5.6. In other words, the inception module increases the representational power of the CNN by taking advantage of multi-level feature extraction.

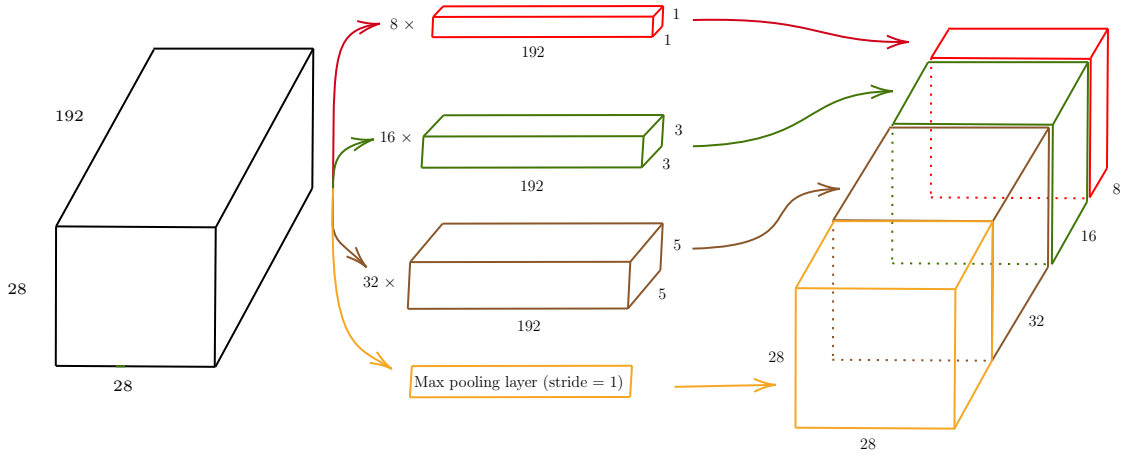


Figure 5.6: Illustration of the naïve inception module. It is noteworthy that a padding is applied to match all the output dimensions.

Nevertheless, a wider network implies a large number of parameters to optimize and a dramatically increasing use of computational resources. To overcome this issue, dimension reduction techniques are required wherever the computational requirements would blow up. Adding 1×1 convolution layer before the expensive 3×3, 5×5 convolutions and after the pooling layer acts as dimension reduction modules, as proposed in [Szegedy et al., 2015b]. Then, an inception network is a network consisting of inception modules (including the dimension reduction technique) stacked on top of each other.

GoogLeNet or Inception-V1 was the first incarnation of the inception architecture [Szegedy et al., 2015b]. It is a network of 22 layers, including nine inception modules. Also, the fully-connected layer was replaced by an average pooling, resulting in better accuracy on ImageNet dataset. Since 22 layers are relatively a large depth, auxiliary classifiers connected to intermediate layers of the network are added to help increasing the gradient signals in the backward pass, thus sidestepping the problem of vanished gradient. They contain a global average pooling layer, then one convolution layer followed by two fully-connected layers. Theoretically, these classi-

fiers encourage the discrimination of the features produced by the middle layers. At training stage, the losses of the auxiliary classifiers are weighted losses added to the total loss of the network, however, they are discarded at inference stage. GoogLeNet has won the 1st place in the ImageNet ILSVRC 2014 classification competition.

Several improvements have been proposed to GoogLeNet, resulting in three better networks: Inception-V2, Inception-V3 and Inception-V4. In Inception-V2, [Ioffe and Szegedy, 2015] introduced the batch normalization technique. It is used to speed up convergence, stabilize the training, and regularize the model. In [Szegedy et al., 2016c], Inception-V3, a variant of Inception-V2 with 48 layers, was proposed. In this version, they introduced the factorization of convolution layers. Convolutions with large spatial filters (5×5 or 7×7 if any) are replaced by convolutions with smaller filters (e.g. 3×3). They also batch-normalized the fully-connected layer of the auxiliary classifiers as well as to the convolution layers of the network.
With the introduction of Tensorflow [Abadi et al., 2016], training complex network architectures becomes simpler because of the memory optimization techniques implemented in it. In order to optimize the training speed, new version of inception architectures has been introduced in [Szegedy et al., 2016a], namely Inception-V4. It is a more uniform simplified architecture with more inception modules than Inception-V3. In other words, they tuned the layer sizes and shed the unnecessary operations while making it deeper and wider than Inception-V3.

5.2.4 Residual Inception Network

In [Szegedy et al., 2016a], the incorporation of inception modules and residual connections were empirically studied. The combination of both concepts was accomplished by adding a shortcut connection to each inception module. In fact, they replaced the filter concatenation stage of the inception module by a residual connection, thus allowing to take advantage of the residual approach while retaining its computational efficiency. They produced two versions of Inception-Resnet architectures (V1 and V2). Inception-Resnet-V1 is Inception-V3 with residual connections, with roughly same computational cost. Inception-Resnet-V2 is Inception-V4 with residual connections. Albeit the inception networks and their counterparts with residual shortcuts achieved very similar results on ImageNet dataset, the employment of residual connections significantly improves the training speed.

5.2.5 Neural Architectural Search Network

Manually designing deep learning models is a daunting task because the search space of the models is extremely large. For this reason, [Zoph and Le, 2016] have introduced the AutoML project in which they automate the design of machine learning architectures. It consists of a neural network that acts as "controller" and propose new network architectures. They apply a reinforcement learning technique [Mnih et al., 2015] where a learner (or agent) must discover the goal by himself through interacting with the environment. The goal of the agent would be to choose its actions in such a way that the reward signal got from the environment is maximized. The structure of the reinforcement learning approach applied to search for network

architectures is illustrated in Fig.5.8. The search space of the controller consists of convolutional architectures with non-linearities, batch normalization and a selection of connections as presented in section 5.2.2 and 5.2.3 (inception modules and residual blocks). The realization of architecture engineering of CNNs often identifies these repeated motifs. Driven by this observation, the controller may have the capacity of predicting a generic convolutional cell expressed in terms of these motifs. This cell can then be stacked in series to construct the full network architecture. Thus, the overall architecture is manually predetermined.

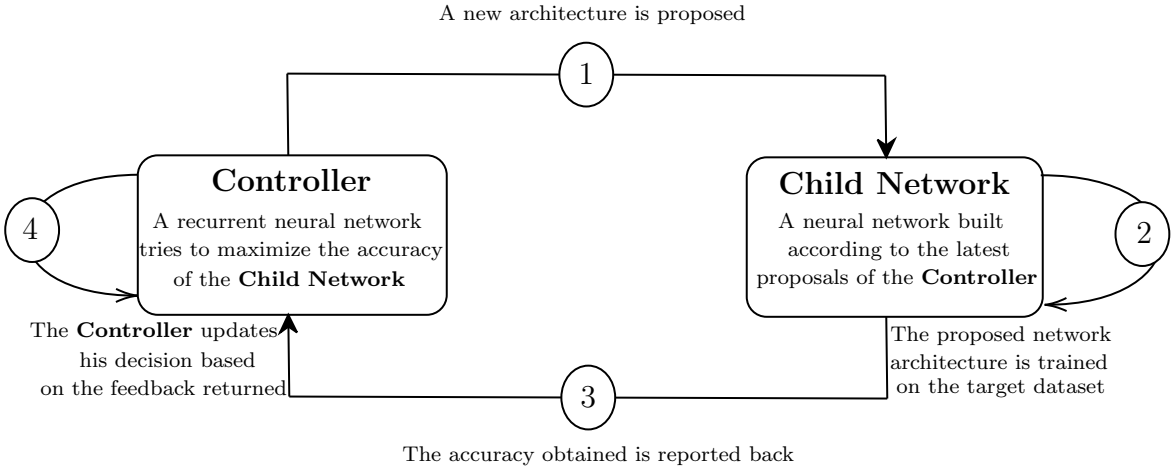


Figure 5.7: An overview of Neural Architecture Search.

Generally, two types of layers are usually required in a CNN: (1) convolution layer that takes a feature map as input and return another feature map. (2) a pooling layer that takes a feature map as input and returns a downsampled feature map. Thus, the controller should produce two types of cells to represent these two operations. As stated in [Zoph et al., 2017], the first type and second type of convolutional cells are referred as Normal Cell and Reduction Cell respectively. This configuration allows to build scalable architectures for images of any size. For more details about those cells, we refer the reader to [Zoph et al., 2017]. The first fruitful results were obtained on Canadian Institute for Advanced Research dataset (CIFAR-10), where a network of three Normal Cells and two Reduction cells are stacked to produce a state-of-the-art performance on this dataset. This network is illustrated on the left of Fig.5.8. The best accuracy on CIFAR-10 is obtained using $N = 7$, with the default input image size 32×32.

Nonetheless, training such approach on a huge dataset as ImageNet is computationally expensive, instead, in [Zoph et al., 2017], they attempted to transfer the knowledge acquired on CIFAR-10 to ImageNet. The best convolutional cells on the CIFAR-10 dataset are applied to the ImageNet dataset by stacking more copies of these cell, each with its own parameters, to produce the NASNet architecture. The best accuracy on ImageNet is obtained by a model called NASNet-A consisting of $N = 7$ and an input image size 331×331. Admittedly, the Network Architectural Search approach has found CNN models better than most human-invented

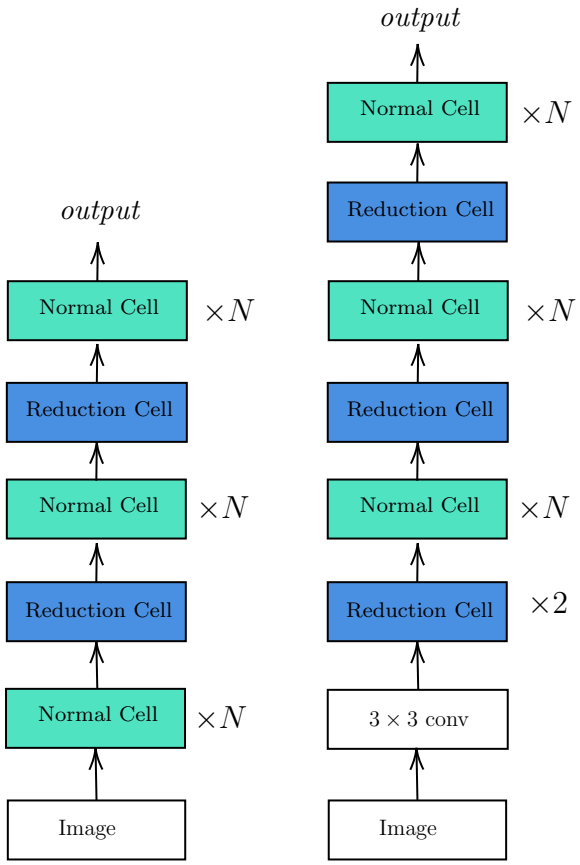


Figure 5.8: Courtsy of [Zoph et al., 2017]. Scalable architectures for image classification task. **Left:** Model architecture for CIFAR-10. **Right:** Model architecture for ImageNet. N is a hyperparmeter to be chosen empirically.

architectures. To the best of our knowledge, NASNet-A is the new state-of-the-art performance on ImageNet classification task.

5.3 Change Detection

As described in section 4.1, the *change* detection is one of the strategies to recognize the tools put on or taken from the surgical tray by detecting only the changes occuring on the tray along the surgery. A *change* occurs whenever a tool is put on or taken from the surgical tray. In other words, we are chiefly interested in telling the difference between a tool put on or taken from the surgical tray and any other task accomplished on the tray including the simple moves of the tools. Recognizing this difference can identify the moments where a tool is probably present in the microscope field of view. In section 4.1, we have shown the results of the *change* detection using a pipeline based on handcrafted and shallow learning visual features. In this section, we propose to automatically learn the discriminative visual features using deep convolutional neural network architectures, such as those described in section 5.2, to address the *change* detection problem.

5.3.1 Model Formulation

In the patch-based solution, the surgeons actions were defined by a couple of images, one before the action starts and another one after the action stops (see section 4.1). This implicitly means that a variable time period was used to define the actions on the tray. On the contrary, a fixed time period γ (in seconds) is used in this study. For each time period γ, two images are extracted to define the actions: one is the first image in the time period, referred as I_1, and the other is the first image in the next time period, referred as I_2. Here, the action is represented by two images $I^a = I_1 - I_2$ and $I^d = I_2 - I_1$, by hypothesizing that I^d is containing the tools taken from the tray (tools disappeared) and I^a is containing the tools put on the tray (tools appeared). More concretely, a video V of s seconds is depicted by $\frac{s}{\gamma}$ couples of (I^d, I^a), which can be expressed as follows:

$$V = \left\{ (I^a, I^d)_1, (I^a, I^d)_2, ..., (I^a, I^d)_{\frac{s}{\gamma}} \right\} \tag{5.13}$$

Thus, a video is described by a list of $2 \times \frac{s}{\gamma}$ images, representing the tools changes (appearances and disappearances) and the other tasks done by the surgeons on the tray. Those images are then fed to a CNN to get a confidence score, which will be used to evaluate the performance of the task. In fact, the network acts as features extractor in its convolutional and pooling layers and it classifies those features using the fully-connected layer(s). In this thesis, the approach of using the image to feed the CNN is referred as I-CNN.

5.3.2 Experimental Setups

In this section, we present the configurations/settings used in order to perform the change detection classification task using deep CNNs architectures. In section 5.3.2.1, we describe the dataset used to perform the task. Afterwards, we present the network parameters and configurations applied to this task in section 5.3.2.2.

5.3.2.1 Dataset

Similar to the dataset used in the patch-based tool presence detection solution, the RW dataset was divided into two subsets: training and test sets, by using the same constraints (see section 4.2.2). The training subset was as well divided into two subsets: learning and validations subsets. In order to be able to optimize the CNN and later the RNN, two complete training videos were assigned to the validation subset. The remaining 23 videos were assigned to the learning subset. The validation videos were chosen such that all tools appear in the learning subset: it was not possible to ensure this property for both subsets.

In this study, we set the time period γ equal to 1 second. Thus, 35450 images where extracted from 23 videos for the learning set. In regards to the ground truth of I^d, we consider the action a tool change whenever a tool is taken from the tray between I_1 and I_2. Idem for I^a, but this time for the tools put on the tray. On the surgical tray, most of the tools only move a few times during a normal conduct of a surgery. By virtue of this property, the images with no tools changes represents

96% of the learning subset, as illustrated in Fig.5.9. Those numbers are subject to change when modifying the time period γ. In Fig.5.10, we present the tools changes frequencies in the three subsets, indicating that the viscoelastic cannula, implant injector, Rycroft cannula and micromanipulator are constantly used in the cataract surgery. Although the frequency of Troutman forceps changes is important, it is rarely used in the surgical field. This is due to the fact that the Troutman forceps is commonly used for preparing the implant.

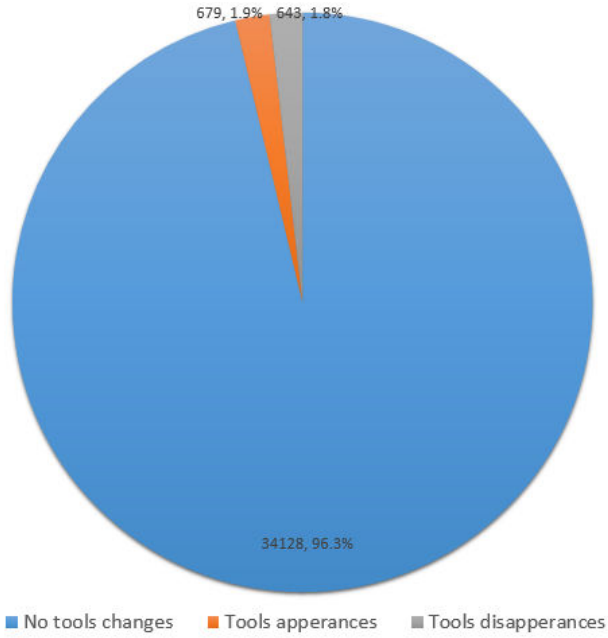


Figure 5.9: Learning subset distribution with a time period $\gamma = 1$. 34128 is the number images with no tools changes, representing 96.3% of the learning subset. 1322 is the number of images with tools changes, which are roughly equally distributed between the tools appearances and disappearances.

For data augmentation purposes, random contrast enhancement, rotation, translation and scaling operations are applied to each image at each training epoch. Also, a global color normalization was applied to each input image.

5.3.2.2 Networks Configurations

The networks used in this study are ResNet-152, Inception-V4 and Inception-ResNet-V2. The full HD definition of the images is considered too high to train these CNN architectures. Images are downsampled to the input image size used for ImageNet: 299×299 for Inception-V4, 299×299 for Inception-ResNet-V2 and 224×224 for ResNet-152. To preserve the aspect ratio, images were first resized to 299×168 for Inception-V4 and Inception-ResNet-V2, and to 224×126 for ResNet-152, then padded with zeros (black pixels) at the top and the bottom to obtain the square images. All CNNs are trained using the RMSProp optimization technique with with momentum 0.9 and a learning rate initialized to $\nu = 0.01$ decaying exponentially for all layers every 2 epochs. The weight decay λ is set to 4×10^{-5}. Here,

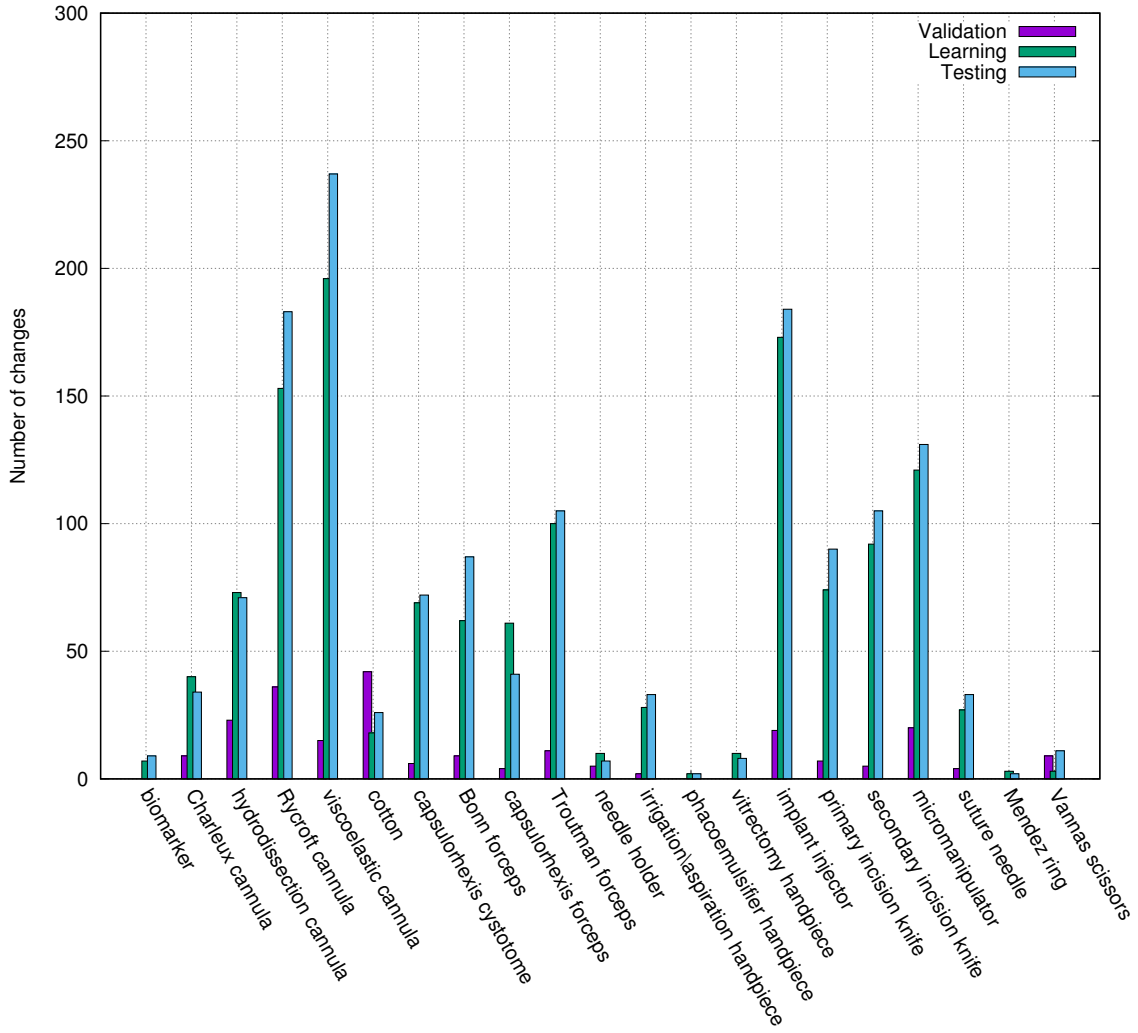


Figure 5.10: Tools change frequency in the learning subset with a time period $\gamma = 1$.

change detection is regarded as a binary classification task. To compute the loss, the cross-entropy function, detailed in Equation (5.4), is used in this context.

Since deep CNN models typically require a large training dataset, we take advantage of transfer learning approaches. The TensorFlow-Slim[1] implementation of these CNNs was trained on a GeForce GTX 1080 Ti GPU, with weights of all the layers pre-trained on ImageNet. The last layer of each CNN, which computes one logit prediction per class, was resized from 1000 neurons for ImageNet to one neuron for change detection; the weight of this neuron was initialized at random.

[1] https://github.com/tensorflow/models/tree/master/research/slim

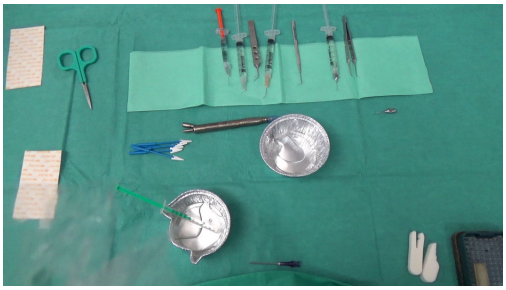
Network architecture	A_z
ResNet-152 (I-CNN)	**0.956**
Inception-V4 (I-CNN)	0.937
Inception-ResNet-V2 (I-CNN)	0.949

Table 5.1: Performance A_z of detecting tools changes in the surgical tray videos. The best result is in bold.

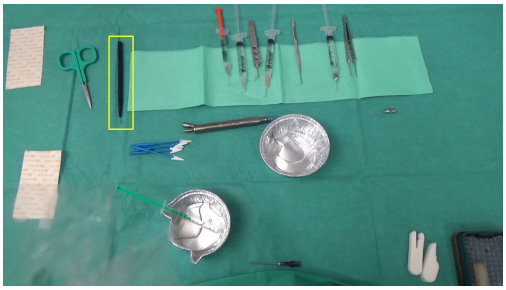
5.3.3 Experimental Results

Image-level classification results of detecting the tools changes are presented in Table 5.1. The area under the ROC curve (A_z) is used as performance metric. The results show that these CNNs are powerful tools to extract the visual features that differentiate between a tool change and any other task done by the surgeons on the tray. Intuitively, the input images are mostly black, as illustrated in Fig.5.12(c)(d), 5.11(c)(d) and 5.13(c)(d). This is due to the pixel subtraction operation applied on the action images and to the small time period ($\gamma = 1$) used in this study. The good results indicate that the CNNs have reaped the benefits of having solely small colored regions of interest in the input images I^a and I^d. The best result is obtained using ResNet-152. One can argue that the network with the largest depth (i.e. 152 layers for ResNet-152) have produced more discriminative visual features than the features produced by the multi-level feature extraction in the inception modules existed in Inception-v4 and Inception-Resnet-V2.

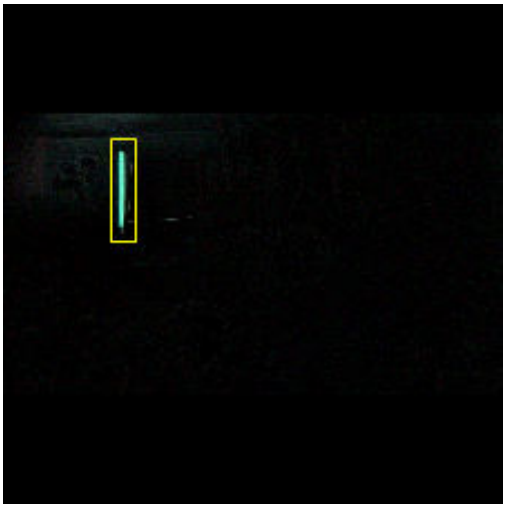
To better understand what the network has learnt, we show in Fig.5.11 (e)(f), 5.12(e)(f) and 5.13 (e) (f) the salient pixels that contribute in the image-level predictions for ResNet-152. These heatmaps are based on the sensitivity criterion: they show the derivative of the tool change prediction with respect to the value of each pixel in the input image [Quellec et al., 2017b]. A green pixel in the heatmap indicates that modifying the corresponding pixel in the input image would change the tool change prediction. In Fig.5.11 and 5.12, straightforward tools changes actions are presented, in which two tools are put on the tray. One can obviously notice the precision of the ResNet-152 in focusing on the targeted tools in Fig.5.11 (e) and 5.12(e). However, in Fig.5.11 (f) and 5.12(f), the salient pixels are scattered which is implicitly indicating that no tools changes are occurring, i.e. nothing has disappeared from the tray. On the down side, the surgeons hands are considerably present in the action images due to the fixed value of γ. In Fig.5.13, we show a complicated tool change action where a tool is taken from the tray while having the intern (surgeon's assistant) preparing the implant. In this action, the salient pixels are dispersed, as illustrated in Fig.5.13(e)(f). This might due to: (1) the surgeon's hands, that cover the targeted tool before taking it, making the tool partially visible or occasionally invisible in I_1 or I_2. (2) the assistant's hands, that prep the implant simultaneously, produces more regions of interest to be addressed by the network.



(a) Sample of I_1

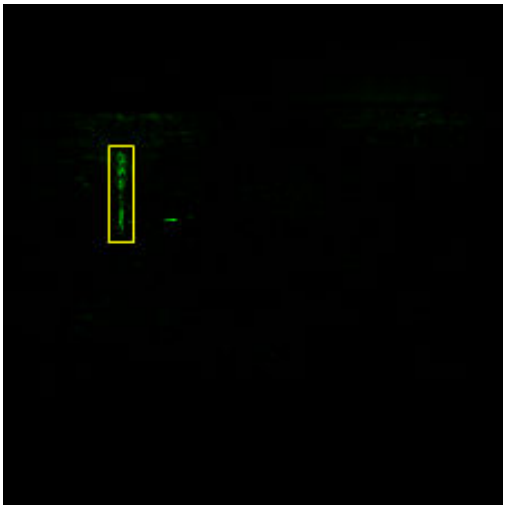


(b) Sample of I_2

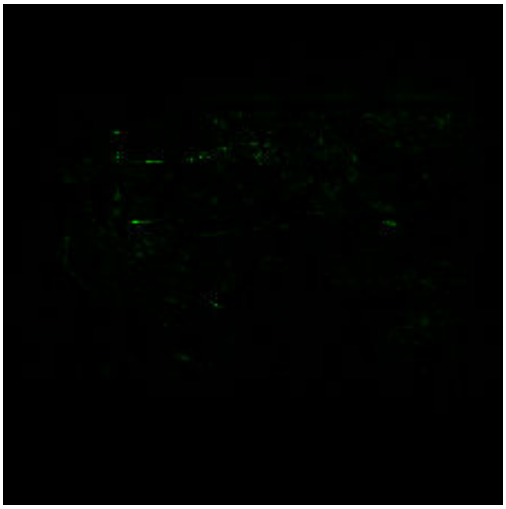


(c) $I_1 - I_2$

(d) $I_2 - I_1$

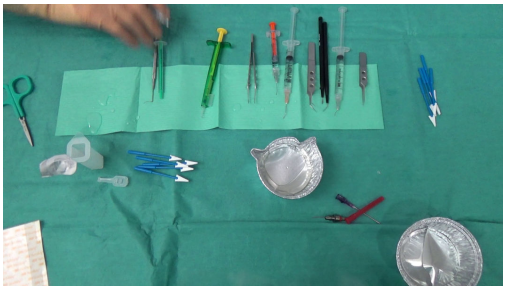


(e) Salient pixels for $I_1 - I_2$

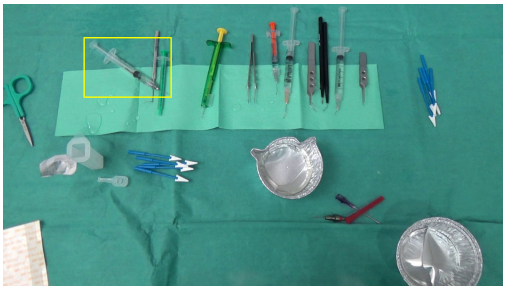


(f) Salient pixels for $I_2 - I_1$

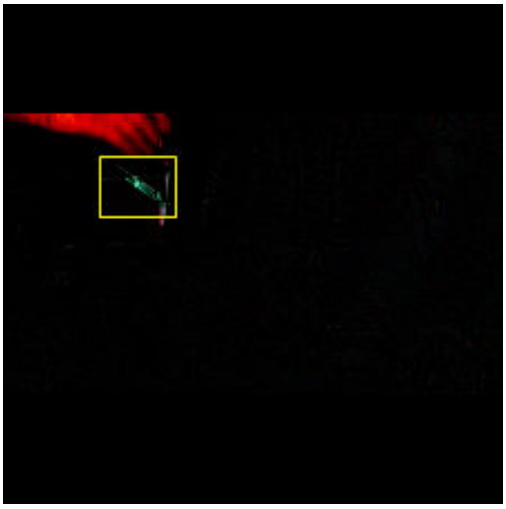
Figure 5.11: Two examples of tools changes. (a) and (b) represents the real scene images of an action with $\gamma = 1$. (c) and (d) are the input images that contain the tools changes. (e) and (f) are the hue-constrained sensitivity analysis for ResNet-152. Yellow boxes contain the tools changes occurred in this action.



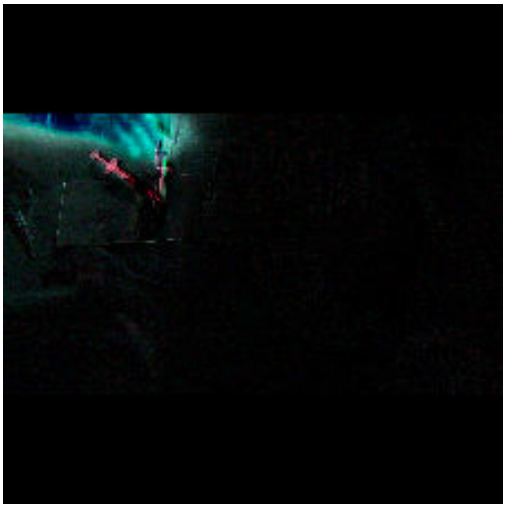
(a) Sample of I_1



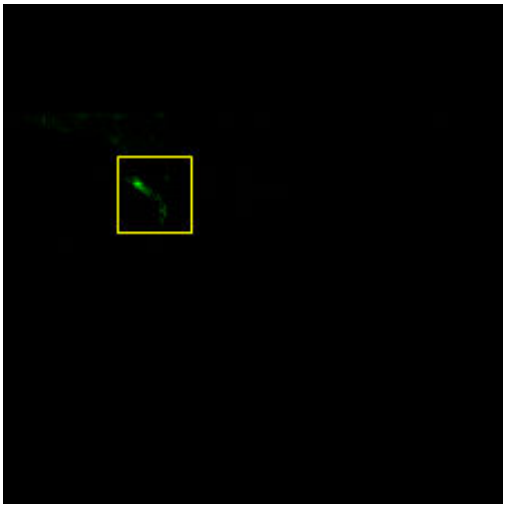
(b) Sample of I_2



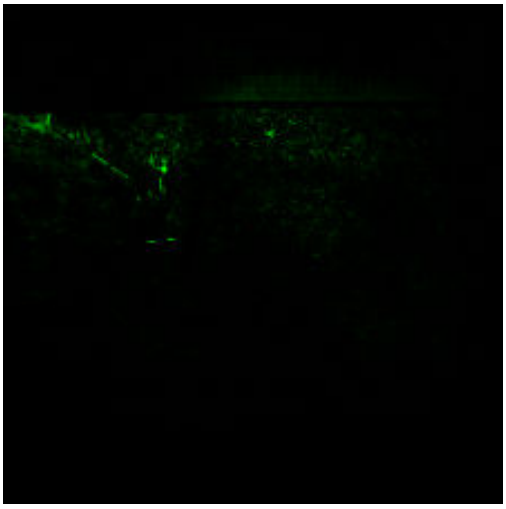
(c) $I_1 - I_2$



(d) $I_2 - I_1$

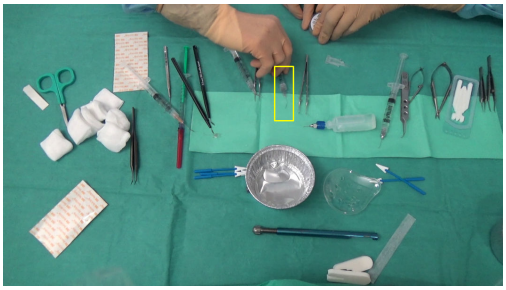


(e) Salient pixels for $I_1 - I_2$

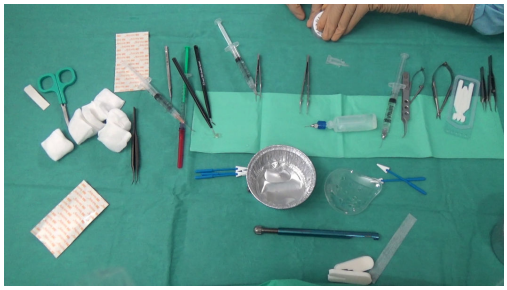


(f) Salient pixels for $I_2 - I_1$

Figure 5.12: Figure 5.11 (Cont.) .

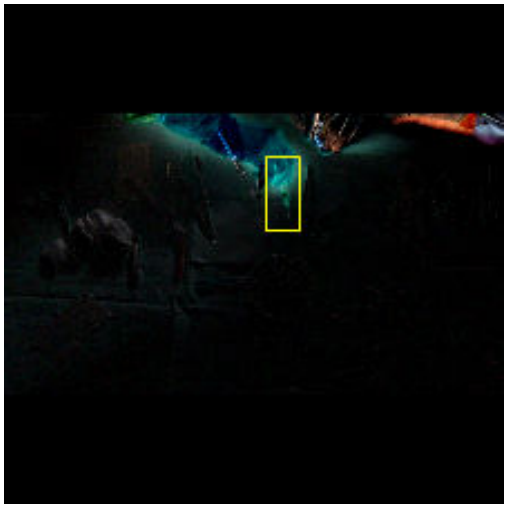


(a) Sample of I_1

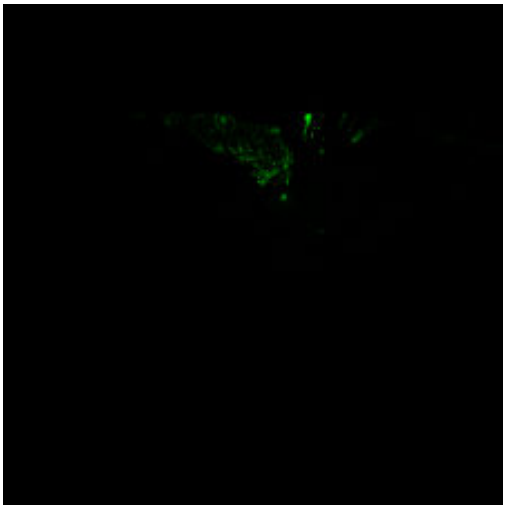


(b) Sample of I_2

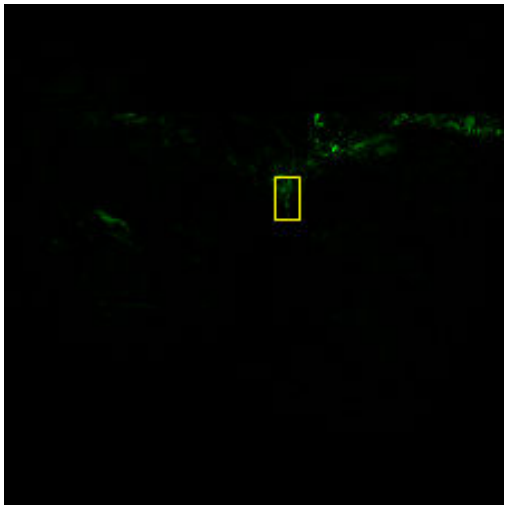
(c) $I_1 - I_2$



(d) $I_2 - I_1$



(e) Salient pixels for $I_1 - I_2$



(f) Salient pixels for $I_2 - I_1$

Figure 5.13: A complicated example of tools changes. (a) and (b) represents the real scene images of an action with $\gamma = 1$. (c) and (d) are the input images that contain the tools changes. (e) and (f) are the hue-constrained sensitivity analysis for ResNet-152. Yellow boxes contain the tools changes occurred in this action.

5.3.4 Change Detection Conclusion

In this study, we have proposed a CNN-based solution for the tools changes detection problem. In fact, we studied the feasibility of the proposed pipeline to detect the tools put on or taken from the surgical tray. This pipeline was based on action images, in which the action is considered a tool change whenever a tool is put on or taken from the tray. Two images representing the action were extracted for each time period γ, then fed to well-known CNNs to get the predictions. Our proposed pipeline shows good performance in performing the task. The experimental results considerably show the efficiency of the proposed method with $A_z = 0.956$ for ResNet-152. These results are marginally inferior to the results obtained using the patch-based approach ($A_z = 0.959$). Nevertheless, the hands are frequently present in the action images, leading to partially or thoroughly covered tools and to more complicated tools changes scenes. This property occasionally dissuades the network from recognizing the targeted tools changes.

5.4 Tool Presence Detection

In section 4.2, we have presented a patch-based approach with shallow learning features to address the problem of tool presence detection in the surgical tray videos. It yielded insufficient results in performing the task. In fact, finding manually the problem-specific discriminative features is a challenging task due to the inherent visual challenges in the tray videos. In this section, we propose to automatically learn the visual features using CNNs. They have the potential to extract better representations from the raw data, leading to much better models. We design multiple well-known CNN architectures to perform surgical tool presence detection on both video types: tool-tissue interaction videos and surgical tray videos. Similar to change detection, we apply the I-CNN approach to perform the task.

5.4.1 Experimental Setups

In this section, we describe the settings followed in order to perform the tool presence detection using deep CNNs architectures for both video types. In section 5.4.1.1, we describe how we split the dataset into three subsets in order to train and evaluate the models. Then, in section 5.4.1.2, we present the network configurations used in this task.

5.4.1.1 Dataset

The RW dataset was divided similarly to the dataset used in the tool changes detection (see section 5.3.2.1). It ends up with three subsets: 23 videos for learning, 2 videos for validation and 25 videos for testing (see Appendix C). The videos were chosen in a manner that all tools appear in the learning and testing subsets: this property was impractical to apply for the validation subset. This division was followed for both video types. Chord diagrams presenting the co-occurrence of tools in the learning tool-tissue interaction videos and the learning surgical tray videos are reported in Fig.5.14 and Fig.5.15, respectively. For further information about the

tools presence in the RW videos, we show the frequency histograms of tools presence (in %) for the tool-tissue interaction videos and the surgical tray videos in Fig.5.17 and Fig.5.16, respectively.

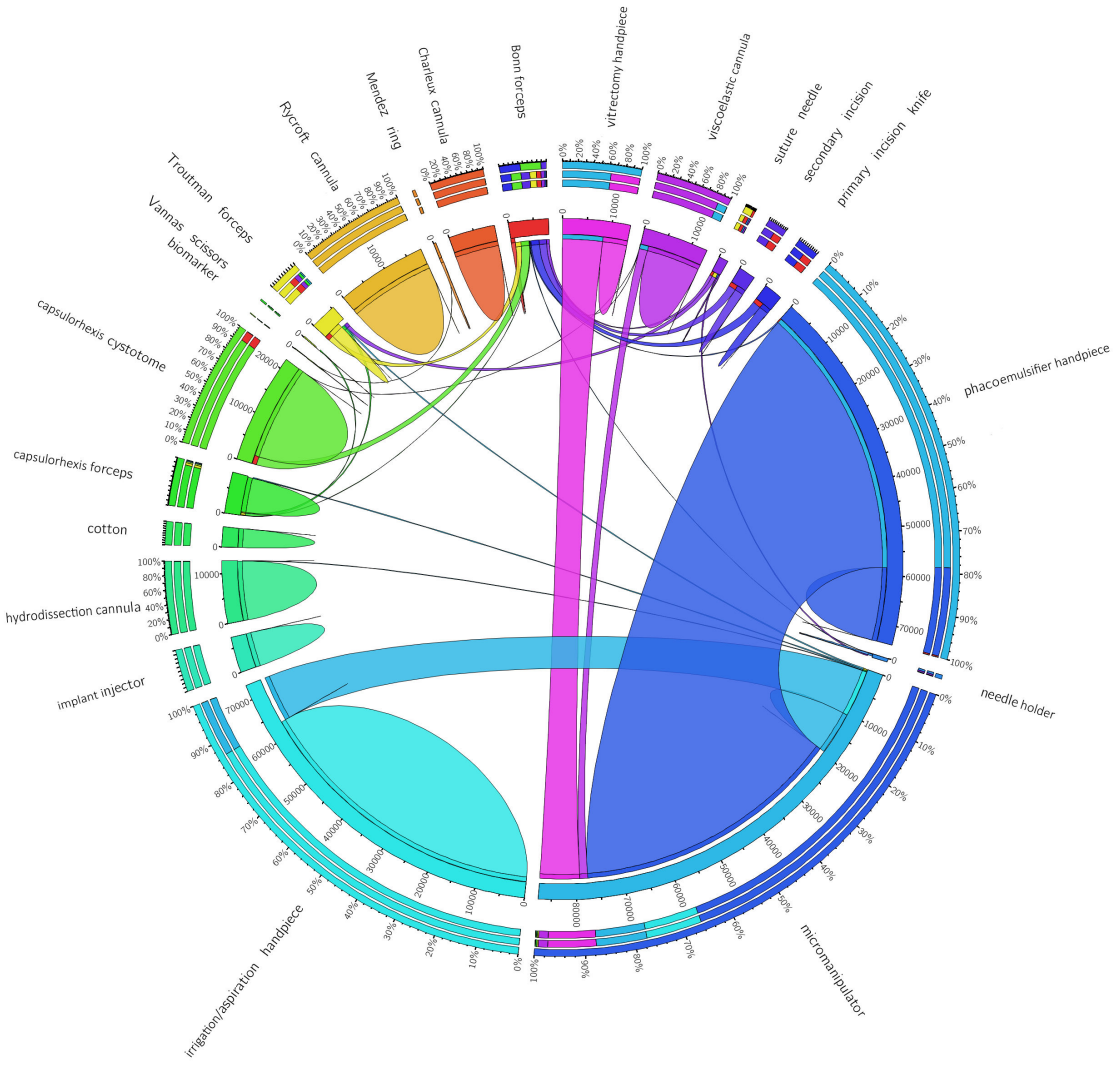


Figure 5.14: Chord diagram illustrating tool co-occurrence in tool-tissue interaction training video frames.

Indeed, one can obviously notice that this dataset has highly unequal tools distribution. In the tool-tissue interaction videos, up to three tools can be used simultaneously. Additionally, the micromanipulator is mostly used with the phacoemulisifier handpiece (see Fig.5.14) and those tools are present in more than 15% of the learning and testing images. However, the biomarker is present in 0.025% and 0.054% of the learning and testing images, respectively, and, it is not present in the validation subset. In addition to the biomarker, the Mendez ring, Vannas scissors and vitrectomy handpiece are not present in the validation subset of the tool-tissue interaction videos. In the surgical tray videos, most of the tools are present concurrently on the tray, as seen in Fig.5.15. The cotton is present approximately in all the images: 99.96 % of the learning images, 99.18 % of the testing images and in all the images

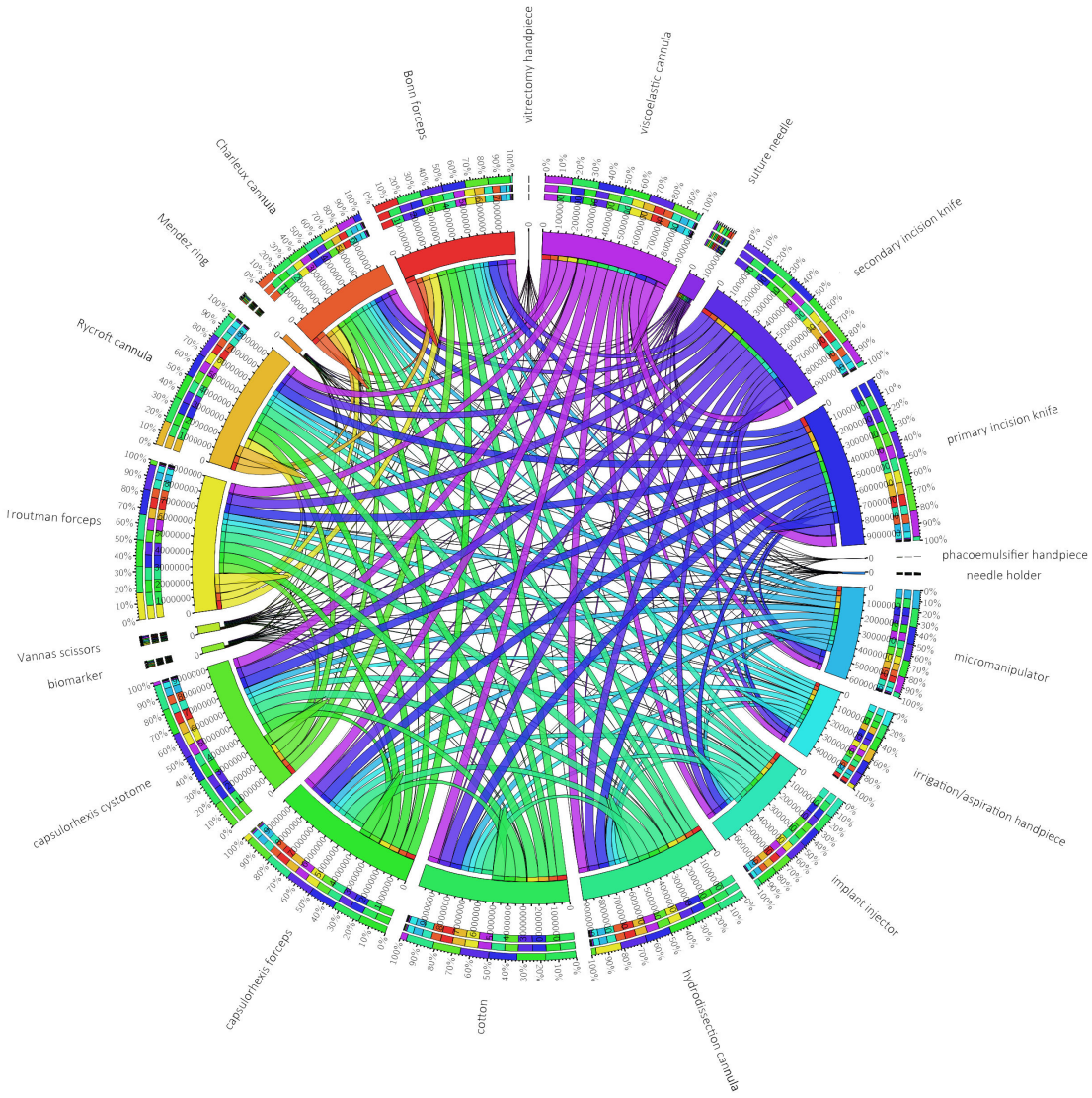


Figure 5.15: Chord diagram illustrating tool co-occurrence in surgical tray training video frames.

of the validation subset. The phacoemulsifier handpiece and vitrectomy handpiece are rarely present on the tray: 0.005 % and 0.003 % for the former, and, 0.32 % and 0.13 % for the latter of the learning and testing subsets, respectively. In addtion to those tools, the biomarker and Mendez ring are not present in the validation subset. A data augmentation approach, similar to the one proposed for the change detection task (see section 5.3.2.1), was applied for each video type.

5.4.1.2 Networks Configurations

For this task, we used the four networks described in section 5.2: ResNet-152, Inception-V4, Inception-ResNet-V2 and NASNet-A. For I-CNN, the input image size is similar to the one used in the change detection problem for Inception-V4, Inception-ResNet-V2 and ResNet152. For NASNet-A, images were resized to 331 ×

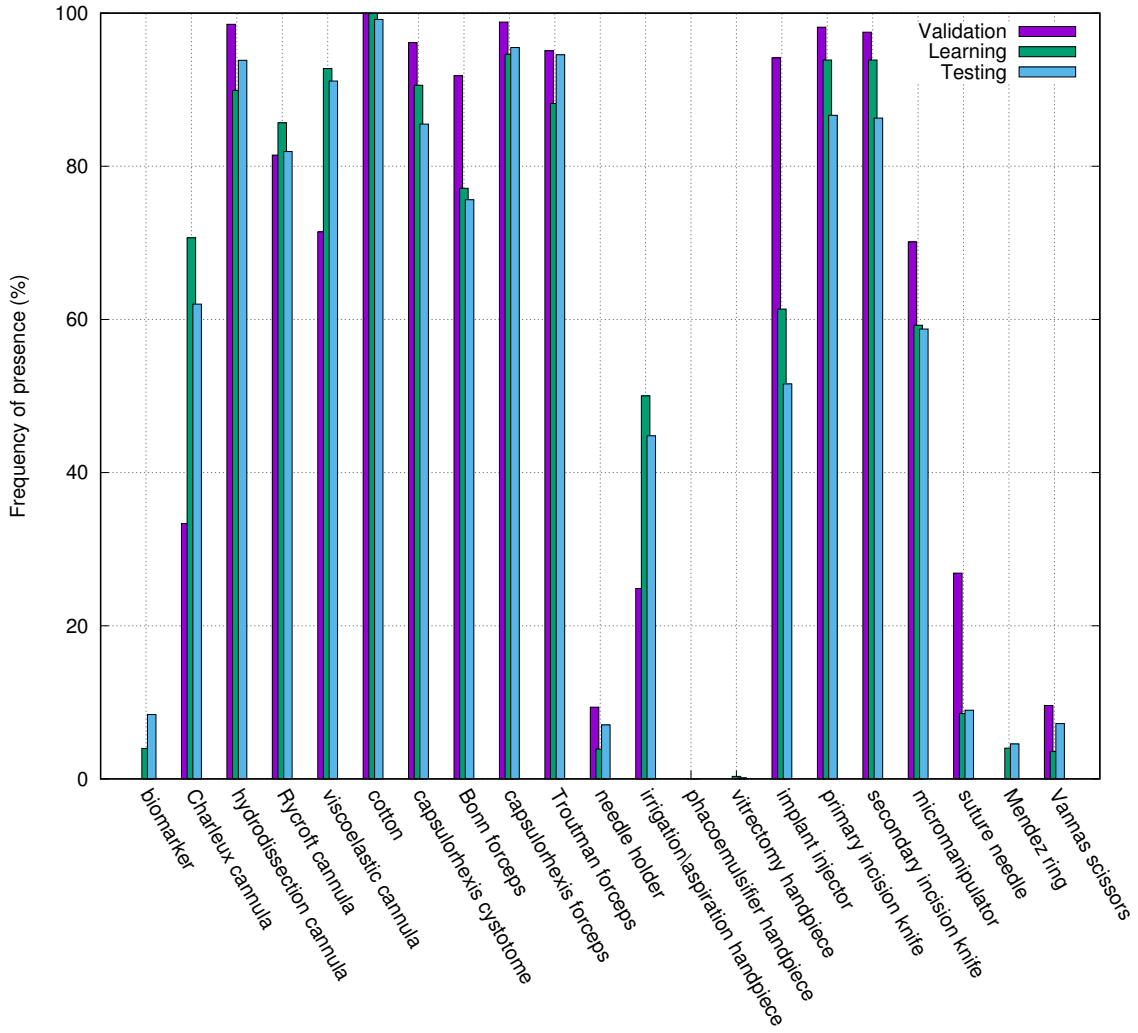


Figure 5.16: Frequency histogram of tool presence in the tray videos subsets.

186. To obtain squares images of 331×331, those images were padded with zeros at the bottom and the top. The learning rate, weight decay and optimization algorithm are identical to the ones used for the change detection problem (see section 5.3.2.2). In this study, tool presence detection is regarded as a multi-label classification task. The cross-entropy function, detailed in Equation (5.6), is used to compute the loss for this task. In addition, a transfer learning design using Tensorflow Slim was applied to this task. It implies the resizing of the output layer of the CNNs from 1000 neurons to 21 neurons (21 tools to classify); their weights were randomly initialized.

5.4.2 Experimental Results

In this section, we report separately the experimentations accomplished on the tool-tissue interaction videos and surgical tray videos using I-CNN. Similar to previous

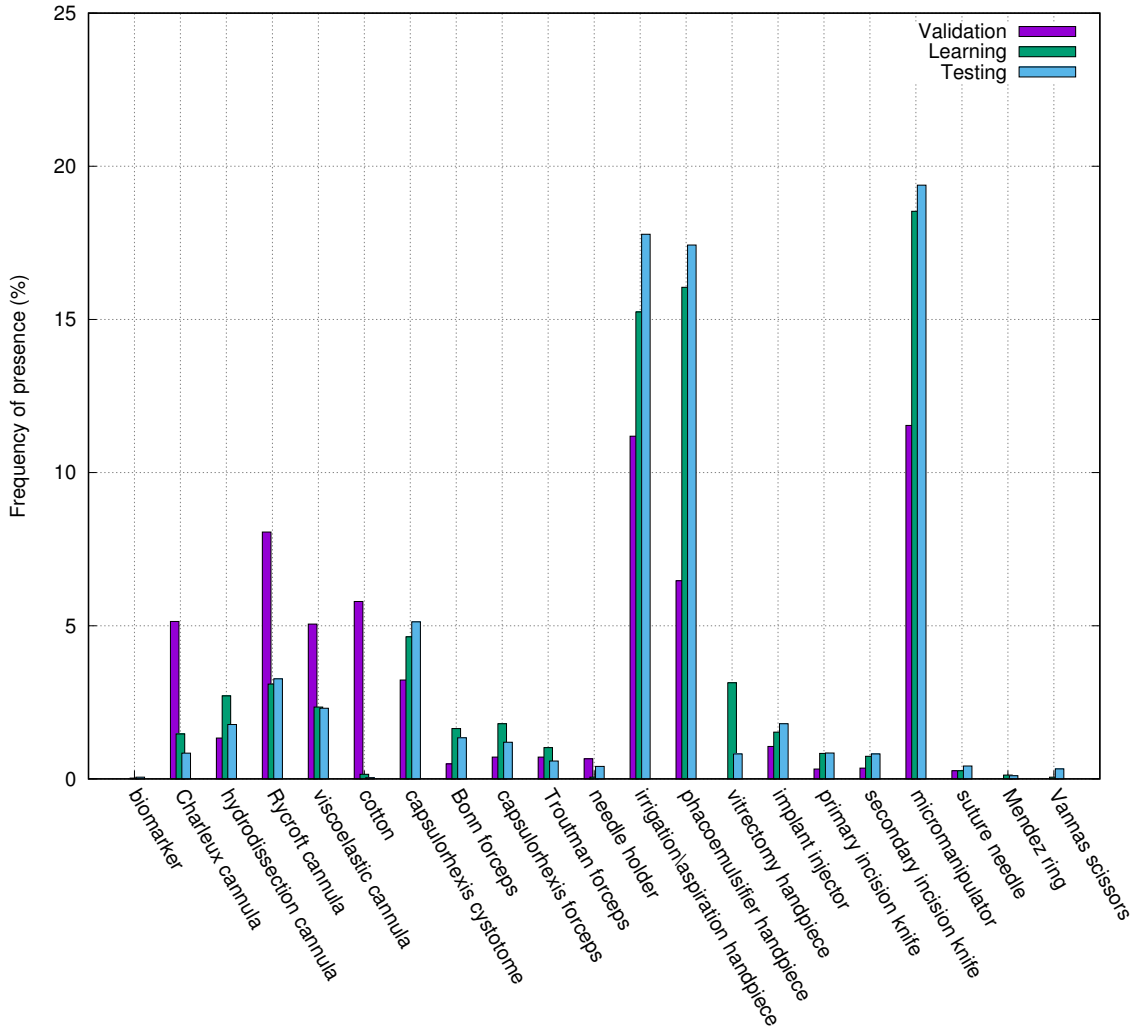


Figure 5.17: Frequency histogram of tool presence in the tool-tissue interaction videos subsets.

experiments, the area under the ROC curve is used as evaluation metric.

5.4.2.1 Tool-Tissue Interaction Videos

In Table 5.2, we report the results of this task using four different networks. One can obviously notice that the CNNs are strongly capable of performing the task in the tool-tissue interaction videos. In particular, the NASNet-A architecture largely outperforms the other models with $mA_z = 0.983$.

Tool	ResNet-152	Inception-V4	InceptionResNet-V2	NASNet-A
biomarker	0.739	0.697	0.891	**0.954**
Charleux cannula	0.899	0.901	0.928	**0.96**
hydrodissection cannula	0.962	0.979	0.961	**0.98**
Rycroft cannula	0.976	0.967	0.976	**0.989**
viscoelastic cannula	0.959	0.945	0.941	**0.962**
cotton	0.911	0.781	0.865	**0.991**
capsulorhexis cystotome	0.994	0.997	0.995	**0.998**
Bonn forceps	0.963	0.963	0.954	**0.98**
capsulorhexis forceps	0.952	0.971	0.938	**0.987**
Troutman forceps	0.968	0.978	0.982	**0.988**
needle holder	0.881	0.823	0.865	**0.991**
irrigation/aspiration handpiece	0.993	0.994	0.99	**0.996**
phacoemulsifier handpiece	0.996	0.997	0.996	**0.998**
vitrectomy handpiece	0.931	**0.981**	0.964	0.957
implant injector	0.978	**0.979**	0.974	0.976
primary incision knife	0.972	**0.989**	0.982	0.981
secondary incision knife	0.968	0.98	0.938	**0.997**
micromanipulator	0.988	0.987	0.991	**0.995**
suture needle	0.963	0.964	**0.985**	0.975
Mendez ring	0.986	0.967	0.876	**0.991**
Vannas scissors	0.853	0.84	0.814	**0.984**
Average (mA_z)	0.945	0.938	0.944	**0.983**
Standard deviation	0.06	0.08	0.05	0.01

Table 5.2: I-CNN results in terms of areas under the ROC curve (A_z) for tool-tissue interaction videos. For each tool, the highest score is marked in bold.

Fig.5.18 reports hue-constrained sensitivity heatmaps for all four CNNs. As can be seen from the examples groundings in this figure, the models discover highly discriminative features, even for relatively rare tools such as biomarker (see Fig.5.18). Those features are scattered over all the input image; they are not only related to the targeted tools but also to the anterior segment of the eye (the lens and the cornea). One possible reason is that each tool interacts differently with the eye, thus analyzing the eye structures assists in differentiating the tools.

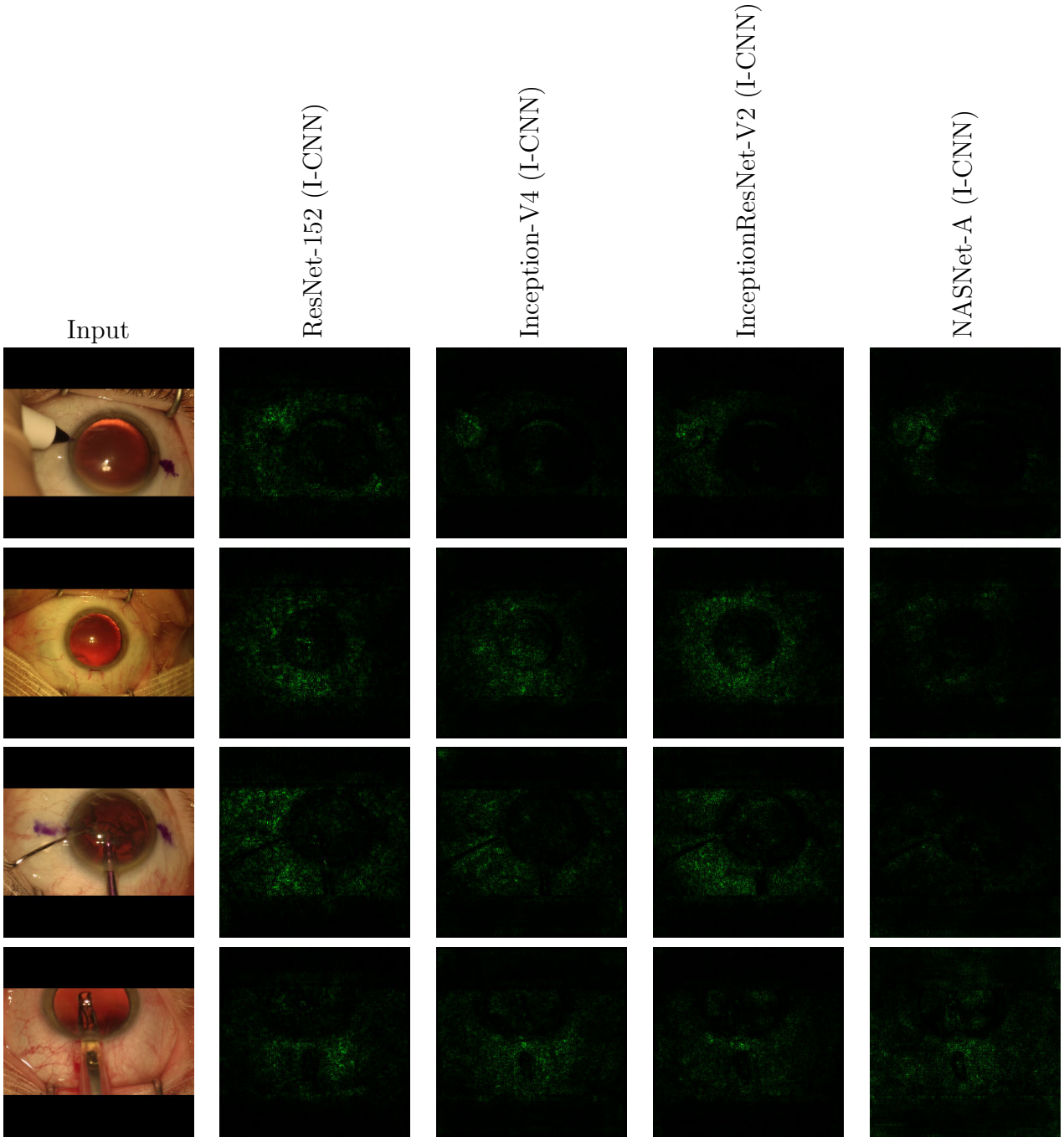


Figure 5.18: Hue-constrained sensitivity analysis for the CNNs. These examples were taken from the testing set of the tool-tissue interaction videos.

5.4.2.2 Surgical Tray Videos

The results of applying I-CNN on the surgical tray videos are reported in Table 5.3. They indicate that the models have badly performed the tool presence detection. The best result is obtained using NASNet-A with $\mathrm{m}A_z = 0.713$. We report hue-constrained sensitivity heatmaps for the best performing network in Fig.5.19. One

can note that the network has extracted features related to the unused objects on the tray (see section 3.3.3) as well as to the targeted tools. Indeed, one can seemingly realize that despite having fairly good results A_z for some tools, it does not imply that the models have performed the task appropriately. It is almost impractical to provide a concrete explanation of such results because it is subject to change from one tool to another. Digging into what the CNNs are learning can provide more profound insights.

The confusion matrices are one of the tools that help in visualizing how well the CNNs features distinguish the tools from one another. In fact, each element (row, col) of the confusion matrix is the probability of the data with true class in row (i.e. ground truth class) that is classified as being in class col (i.e. predicted class). In this thesis, we are chiefly interested in detecting tool presence in the operative field and this implies detecting the tools absence on the tray. Thus, detecting tools absence on the tray is as much relevant as the tool presence detection. The tool absence probability can be expressed as $1 - y$, where y is the probability of the tool presence. Yet, surgical tool absence/presence detection is a multi-label classification task, for which the confusion matrix is not applicable. For each element (row, col), discarding the images where the classes in row and in col are simultaneously absent can alleviate this limitation. The confusion matrix for absence detection is illustrated in Fig.5.20.

In Table 5.4, we interpret the confusion matrix according to the tools distribution in the learning subset. The models have poorly detected the presence of all rare tools, such as the biomarker, the phacoemulsifier handpiece, the vitrectomy handpiece, just to name a few. This is also the case for the cotton which is approximately present in all learning images. In fact, the misclassification rates for these tools are primarily affected by the tools presence frequencies in the learning subset. Such mismatched tools distributions impede the network from performing the task properly; it would consistently consider the tool present or not present without attempting to find the appropriate patterns. The relatively poor performance did not preclude the networks from producing very good results for one tool: the viscoelastic cannula with $A_z = 0.95$. One possible reason is the specification of this tool, i.e. the color information is thoroughly different from other tools because the viscoelastic cannula is always mounted on an orange syringe plunger (see Fig.3.5b).

It is noteworthy that the networks yield moderate performance for the most common tools used in the cataract surgery (cannulea, knifes, forceps, etc.), which are simultaneously present in large part of the tray dataset. One can reasonably argue that this property hinders the models from efficiently differentiating the tools, especially the high-similar ones. As illustrated in Fig.5.20, when the capsulorhexis forceps is absent, the probability of having the Troutman forceps absent is 0.156, which is greater than the probability of absence of Troutman forceps (0.119). Similarly, when the secondary incision knife is absent, the probability of having the primary incision knife absent is 0.173, which is greater than the probability of absence of the primary incision knife (0.157). In addition, the Troutman forceps is commonly used for preparing the implant, which is a step occurring after taking the irrigation/aspiration handpiece from the tray (i.e. this tool rarely gets back to the tray). The models have learnt that when the Troutman forceps is absent, the

irrigation/aspiration handpiece is predominantly absent, by having the probability of this case equal to 0.915, as shown in Fig.5.20. This indicates that the models are occasionally learning the co-occurrence of the tools more than differentiating them.

Furthermore, the CNNs used in this study can greatly handle relatively small image sizes, while being susceptible to error-prone when using big image sizes. Most of the tools are small and highly similar to one another (see section 3.3.3). Then, downsampling the video images to the default input image size of the CNNs makes the recognition of those tools by the human eye challenging. One can intuitively argue that the models have similarly suffered in the tools discrimination task.

5.4.3 Tool Presence Detection Conclusion

In this study, we have proposed a list of models to detect the surgical tool presence on the tool-tissue interaction and surgical tray videos. A typical CNN design (I-CNN) was followed for both video types. Quantitative experiments demonstrate that the models, notably NASNet-A, can perfectly perform the task on the tool-tissue interaction videos. The experiments show that the models overcome the inherent challenges in the surgical field videos and subsequently yield discriminative visual features to effectively detect the tools. Interestingly, these features are extracted from the lens and the cornea as well as from the tools. Although the results are very good on the tool-tissue interaction videos, the models have deficiently performed the task on the surgical tray videos. They are subject to multiple limitations. The first limitation is the dataset tools distributions. Having such imbalanced dataset compromises the CNNs performance in the tool presence detection task. In addition, a resampling technique would scarcely amend the tools distribution. Since most of the tools are concurrently present, upsampling the rare tools upsamples the other tools and downsampling the most frequent tools downsamples the rare tools. The performance was fairly good for the most common tools used in the surgery and considerably better than the results obtained using the patch-based solution $\mathrm{m}A_z = 0.6$ (see section 4.2.4). However, the confusion matrix have shown sensible properties of the learnt features representations. The models have occasionally learnt the co-occurrences of the tools in addition to some simple tool-specific motifs (such as the color for the viscoelastic cannula). In view of the complexity of finding small and thin tools in small images, the models have conceptually suffered in the task of discriminating the tools because of the relatively small input image sizes.

According to the objective of this thesis, the pretty good results obtained on the tool-tissue interaction videos are apparently sufficient. Notwithstanding the deficient results obtained on the tray videos, mitigating the challenges, previously discussed, can predominately produce better results on the tray, subsequently having the ability to probably improve the accuracy of detecting the tools in the surgical field.

Tool	ResNet-152 (I-CNN)	Inception-V4 (I-CNN)	InceptionResNet-V2 (I-CNN)	NASNet-A (I-CNN)
biomarker	0.456	<u>0.724</u>	0.676	0.679
Charleux cannula	0.576	0.478	<u>0.67</u>	0.437
hydrodissection cannula	0.759	0.737	0.815	<u>0.827</u>
Rycroft cannula	0.663	0.589	<u>0.758</u>	0.747
viscoelastic cannula	0.87	0.867	0.893	<u>0.95</u>
cotton	0.561	0.375	0.396	<u>0.772</u>
capsulorhexis cystotome	0.79	0.826	<u>0.852</u>	0.75
Bonn forceps	0.808	0.81	0.831	<u>0.864</u>
capsulorhexis forceps	<u>0.834</u>	0.606	0.778	0.768
Troutman forceps	0.601	0.637	0.662	<u>0.709</u>
needle holder	0.467	0.553	<u>0.63</u>	0.599
irrigation/aspiration handpiece	0.844	0.76	0.923	<u>0.932</u>
phacoemulsifier handpiece	0.369	<u>0.874</u>	0.38	0.663
vitrectomy handpiece	0.468	0.247	<u>0.505</u>	0.308
implant injector	0.702	0.728	0.747	<u>0.768</u>
primary incision knife	0.77	0.778	0.798	<u>0.818</u>
secondary incision knife	<u>0.841</u>	0.772	0.811	0.761
micromanipulator	0.672	0.74	0.794	<u>0.856</u>
suture needle	0.408	0.461	0.471	<u>0.701</u>
Mendez ring	0.432	<u>0.562</u>	0.486	0.437
Vannas scissors	0.51	0.571	0.616	<u>0.633</u>
Average (mA_z)	0.638	0.652	0.69	<u>0.713</u>
Standard deviation	0.166	0.166	0.162	0.162

Table 5.3: I-CNN results in terms of areas under the ROC curve (A_z) for surgical tray videos. For each tool, the highest score is underlined.

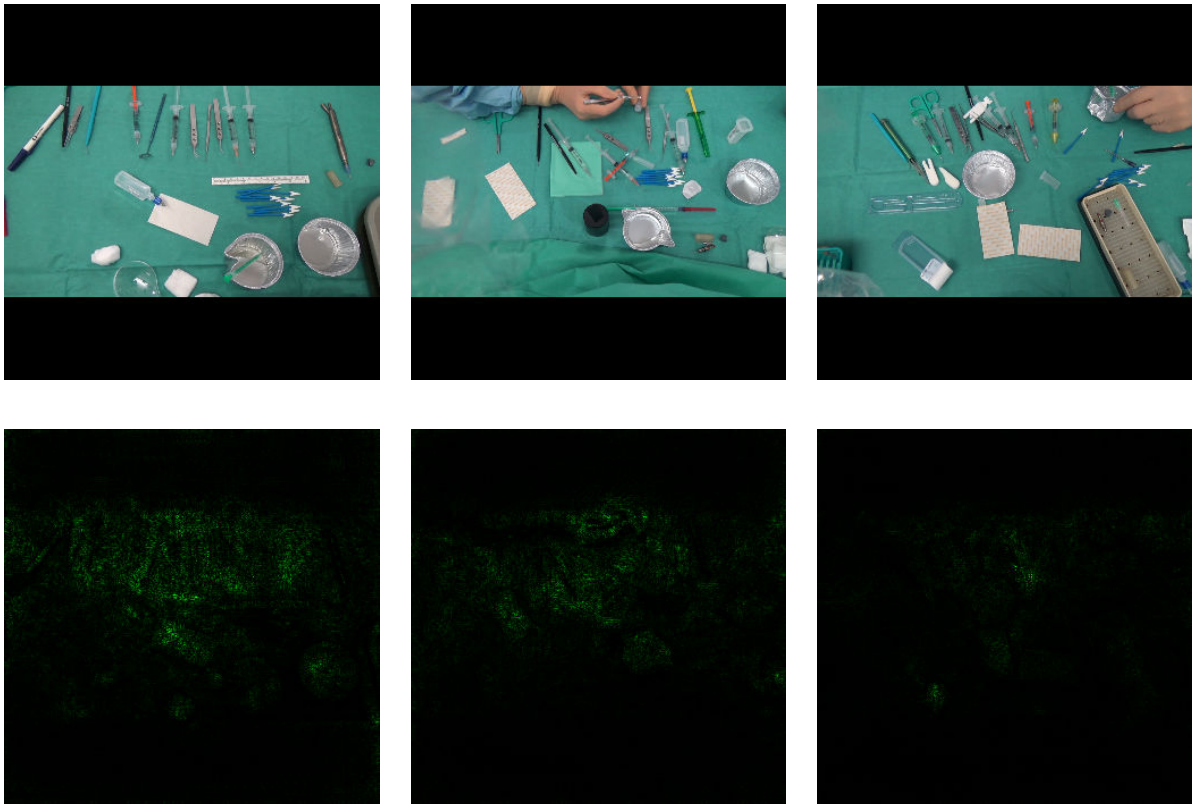


Figure 5.19: Hue-constrained sensitivity analysis for best performing I-CNN: NASNet-A. These examples were taken from the testing set of the surgical tray videos.

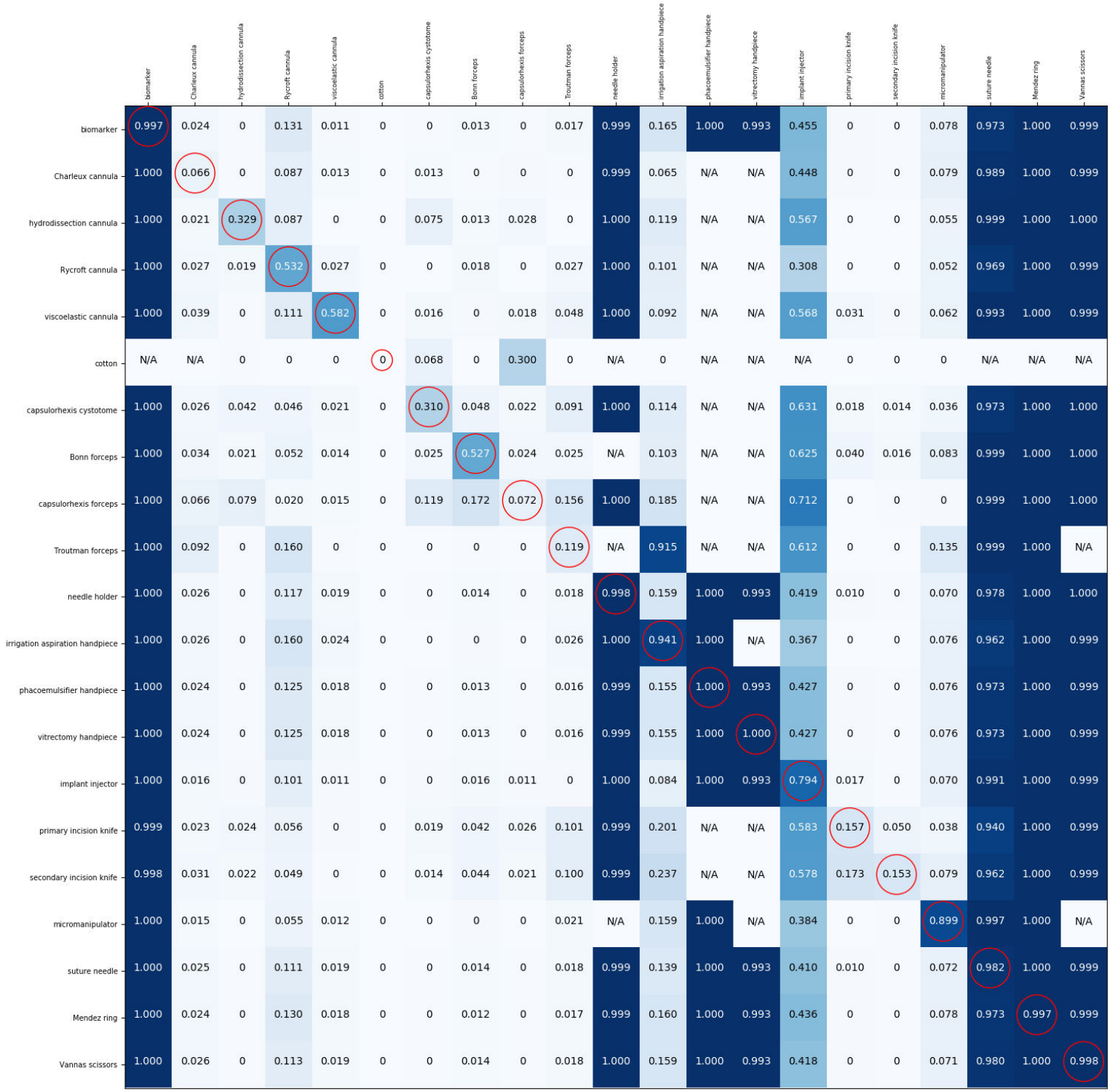


Figure 5.20: Confusion matrix for NASNet-A (I-CNN) tool absence (no presence) detection. For easier understanding, the diagonal cells are circled in red. N/A is not applicable: no images were found where the class in row is absent and the class in column is present.

Tool	Frequency of presence(in %)	Interpretations
phacoemulsifier handpiece	0.005	In Fig.5.20, these tools are always detected as absent on the tray. This interpretation is highly correlated to the frequency of their presence in the learning subset; they are all present in less than 9% of the images, thus the poor performance in Table 5.3.
vitrectomy handpiece	0.323	
Vannas scissors	3.558	
needle holder	3.883	
biomarker	3.983	
Mendez ring	4.008	
suture needle	8.541	
irrigation/aspiration handpiece	50.013	Charleux cannula is the one having the least performance among this list of tools. The probabilities in the confusion matrix showcase the incapacity of the models to differentiate between its absence and its presence. The remaining tools are the most common tools used in the cataract surgery, which are present in a large part of the dataset. The misclassification rates for those tools are significant in the confusion matrix (except for the viscoelastic cannula) even though they achieve fairly good results in terms of A_z.
micromanipulator	59.24	
implant injector	61.343	
Charleux cannula	70.667	
Bonn forceps	77.122	
Rycroft cannula	85.678	
Troutman forceps	88.205	
hydrodissection cannula	89.908	
capsulorhexis cystotome	90.589	
viscoelastic cannula	92.763	
primary incision knife	93.872	
secondary incision knife	93.873	
capsulorhexis forceps	94.62	
cotton	99.969	As seen in the confusion matrix, this tool is always detected as present on the tray. This is probably due to its frequency of presence in the learning set.

Table 5.4: Confusion matrix interpretation according to the tools distribution in the learning subset. Tools are presented in ascending order of their frequency distribution.

5.5 Proposed Solution For Surgical Tray Challenges

We have previously shown that performing the surgical tool presence detection using CNNs on the surgical tray videos is a highly challenging task. In this section, we

propose a solution for these challenges in two folds: first, datasets of artificial tray videos are generated, and, second, a patch-based approach to train the CNNs.

5.5.1 Simulated Dataset

The RW dataset has a highly unequal tools distribution for which a resampling technique was infeasible. In order to alleviate this problem, we propose to generate artificial video datasets that mimic the real-world environment. They consists of realistic surgical tray scenes in which the disposable tools (see section 3.3.1) are the only tools used to record the videos. They are filmed based on carefully predefined and highly specific descriptions that help in addressing the aforementioned problems. Reducing the cost and time for collecting and labelling videos are the main advantages for synthetically generated datasets.

Theoretically, examining the possibility of augmenting the real world tray data with artificially created data is seemingly intriguing. This can be technically expressed as training the models using the synthetic data then finetuning them using the real world data. Despite the efficiency of such approach, it can impede the scalability of the models in this case because the inherent challenges of the RW tray videos are still going to be learnt by the models. We approach the problem differently: the RW data are solely used to evaluate the models. Here, we primarily study the feasibility of training CNN models on simulated data and evaluating them using RW tray data; secondarily, the surgical tool presence detection is also validated using simulated videos. To best of our knowledge, this is the first attempt in the literature to use synthetic data in order to boost the performance of a real world surgical tool detection problem.

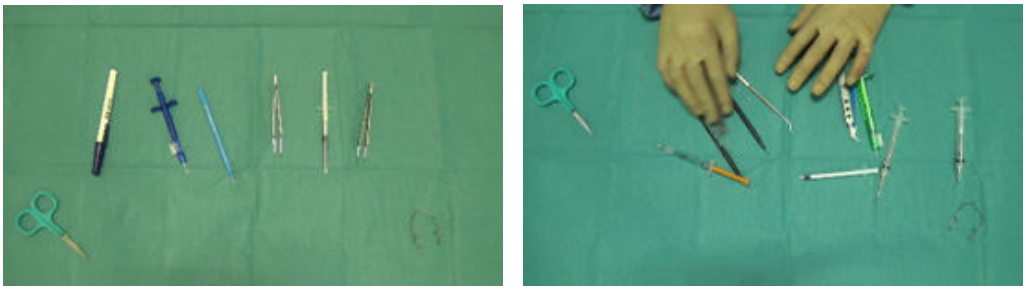
5.5.1.1 Video Setups

A similar setup to RW surgical tray were used to record the artificial videos. The same video camera was attached to the same arm covered by a surgical tray drape. This articulated arm was fixed using a clamp on a table on which the tools are laid down. In the artificial videos, we replicated the same kind of gestures done by the surgeons and the scrub nurses during the surgery, for instance moving around the tools over the tray and taking apart the cannulea from the syringes, etc. In addition, we continually move the tools to get many views as possible of each one. In regards to the ground truth acquisition, the tools are never put on or taken from the scene during a simulated video. Thus, the annotation of the tools can be done at video level, whereas tool presence must be annotated at the image level in the RW dataset, and this is a highly consuming task.

5.5.1.2 Random Number Tools Dataset

We have performed significant experimentations to elucidate which dataset design is the most appropriate. In this thesis, we present the most effective dataset design: Random Number Tools dataset, referred as RNT dataset. The RNT dataset contains short video clips, where each video contains a variable number x of tools selected randomly (see Fig.5.21). The tools were distributed over the videos in a way that

most of them are present in half of the videos. Three videos were recorded for each value of x between 5 and 12, resulting in a dataset of 21 videos of simulated surgical tray scenes. RNT videos had a duration of 3 minutes and 7 s on average (minimum: 2 minutes 11 s, maximum: 4 minutes 48 s). The frame definition was 1920x1080 pixels (full HD resolution) for all the videos. The frame rate was 50 frames per second.



(a) Sample with 5 tools

(b) Sample with 8 tools

Figure 5.21: Samples extracted from the RNT simulated dataset.

5.5.2 Model Formulation

In accordance with the patch-based approach proposed in the previous chapter (see section 4.2.1), we propose to use the image patches to feed the CNNs rather than the whole image, in order to exploit deeply the full HD image of the cataract dataset. For a video V of q frames, each frame I is divided in K patches P. Thus, V can be represented by:

$$V = \left\{ \left\{ P_1, ..., P_K \right\}^{(1)}, \left\{ P_1, ..., P_K \right\}^{(2)}, ..., \left\{ P_1, ..., P_K \right\}^{(q)} \right\} \tag{5.14}$$

For the forward pass, we feed the CNN the list of patches $\{P_1, ..., P_K\}$ instead of the whole image I. With m neurons (classes) at the output layer, the patch P_j has m scores, which can be expressed as a list of scores $S_j = \{s_{j1}, s_{j2}, ..., s_{jm}\}, j \in \{1, ..., K\}$. Then, we use the scores $S = \{\widehat{s}_1, \widehat{s}_2, ..., \widehat{s}_m\}$ of the image I to compute the loss. The score $\widehat{s}_k$ where $k \in \{1, ..., m\}$ of I can be formulated as:

$$\widehat{s}_k = \max(s_{1k}, s_{2k}, ..., s_{jk}) \tag{5.15}$$

In other words, for each class, the maximum score value obtained among the patches is retained. In this thesis, we refer to the patch-based convolutional neural network as P-CNN.

By treating the patches, we improve the quality of images fed to the CNNs, expectedly leading to better performance in differentiating the tools, in particular the high similar ones. In addition, the patches enforce the decoupling of the tools present on the tray, which is conceivably helping the models to focus on finding tool-specific patterns rather than simple tools co-occurrence assumptions.

5.5.3 Experimental Setups

5.5.3.1 Datasets

The RNT dataset was divided in two subsets: 17 videos for learning, 4 videos for validation. The videos were chosen in a manner that all tools appear in the learning and validation subsets: there were no synthetic testing subset. For further information about the tools presence in the RNT videos, we show in Fig.5.22 the frequency histograms of tools presence (in %) for the RNT videos. All tools are present in more than 45% of the learning images, with two of them exceed 85%. Compared to the RW dataset, this distribution is much more balanced. A data augmentation approach, similar to what has been proposed in the previous sections (see section 5.3.2.1), was applied for the simulated dataset.

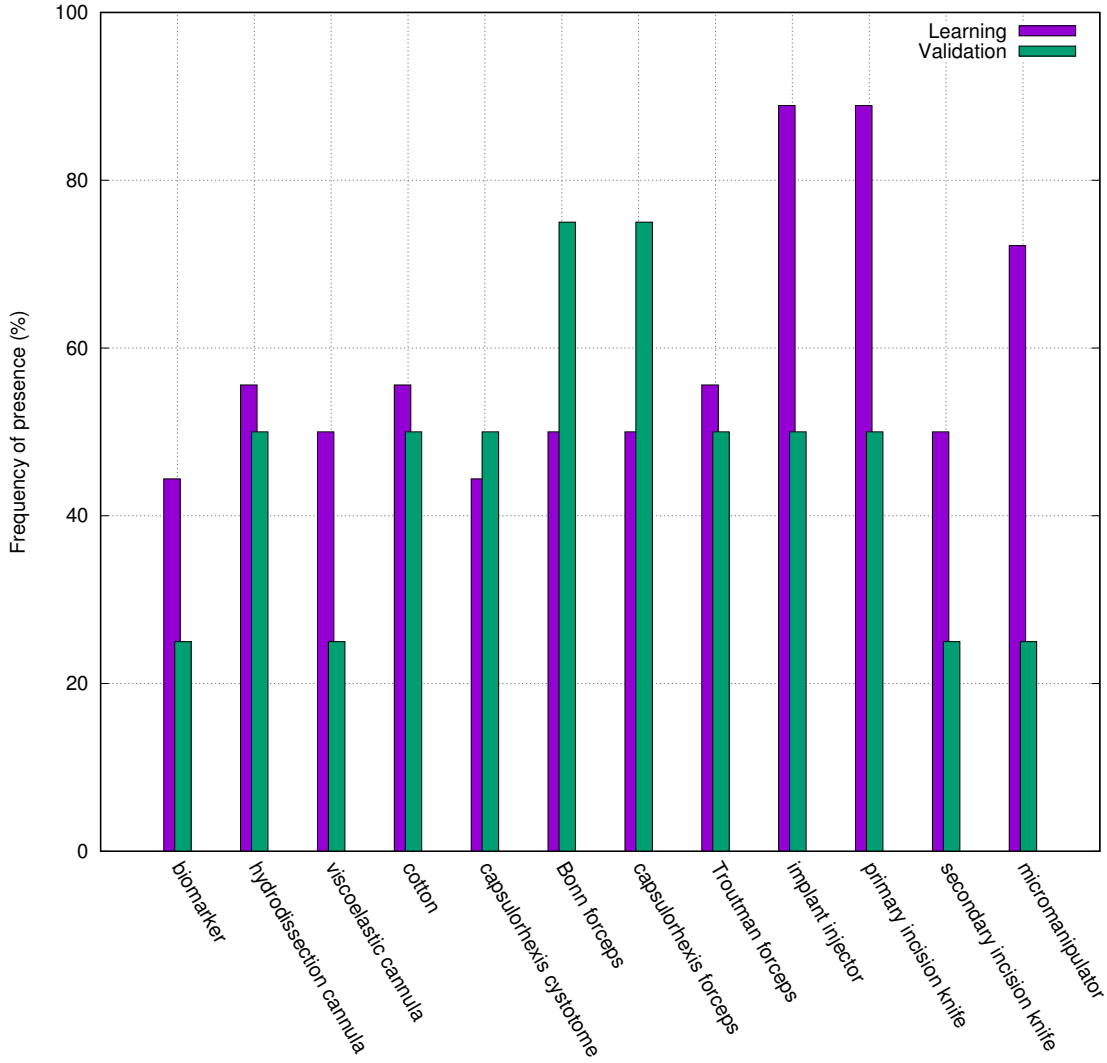


Figure 5.22: Frequency histogram of tool presence in the RNT videos subsets.

5.5.3.2 Networks Configurations

For this task, we used the three networks: ResNet-152, Inception-V4, Inception-ResNet-V2 (see section 5.2). Similar settings to I-CNN (see section 5.3.2.2) are proposed to P-CNN with the exception of the input image settings. The full HD images are first resized to 1063×598. This size maintains the recognition of all the tools by the human eye on the tray (including the needles and cannulas). Then, the images were divided into $K = 11$ patches of fixed size. The patch is a square image of size 299×299 for Inception-V4, 299×299 for Inception-ResNet-V2 and 224×224 for ResNet152. The division was performed in such a way that the images are first split in 8 non-overlapping square patches (i.e. by padding with zeros when appropriate), then three more patches are extracted, spreading from the center point to both bounds of the image.

The learning rate, weight decay, optimization algorithm and input image sizes are identical to those used for the change detection problem (see section 5.3.2.2). Since multiple tools can be present concurrently in a surgical tray scene, tool presence detection is regarded as a multi-label classification task. Equation (5.6) is the cross-entropy function used to compute the loss in this study. Similar to tool detection detection on RW tray videos, Tensorflow Slim was used to apply a transfer learning design: the output layer of the CNNs was resized from 1000 neurons to r neurons, which were initialized randomly. r is the number of tools to be classified; equal to 12 in the case of simulated dataset.

5.5.4 Experimental Results

We report the results of the P-CNN and I-CNN approaches in Table 5.5 and Table 5.6, respectively, on the validation subset as well as to the RW testing subset. One can obviously notice that the CNNs have deficiently performed the task on the simulated data, and subsequently on the RW data. Here, we focus the discussion on the P-CNN approach. The best results for P-CNN was obtained using Inception-ResNet-V2 with $\mathrm{m}A_z = 0.712$ and $\mathrm{m}A_z = 0.643$ for synthetic data and RW data, respectively. Similar to section 5.4.2.2, we compute the confusion matrices to evaluate the accuracy of the classification using the validation subset and the RW testing subset. Despite the main interest of evaluating the RW data on models trained on simulated data, we only discuss the confusion matrix of the simulated data, which is illustrated in Fig.5.23. The confusion matrix of the evaluation using RW data is reported in Appendix B, Fig.9.3.

The biomarker, viscoelastic cannula, cotton, and implant injector are reasonably well detected. Their misclassification rates in the confusion matrices are far from being impactful on the ability to detect their presence.
As for hydrodissection cannula, Bonn forceps, capsulorhexis forceps and secondary knife incision, the confusion matrix indicates that they are mostly considered as absent, thus the poor performance. The Troutman forceps is well detected, however, it is also considered absent when the capsulorhexis cystotome and Troutman forceps are absent. For these cases, one can possibly argue that the CNNs have searched for conjunctions of tools that are easy to detect, which are not necessarily the targeted tools. In other words, they have learnt the tools co-occurrences rather than finding

	RNT validation susbet			RW testing susbet		
Tool	ResNet-152	Inception-V4	Inception-ResNet-V2	ResNet-152	Inception-V4	Inception-ResNet-V2
biomarker	0.896	0.539	0.997	0.487	**0.867**	0.713
hydrodissection cannula	0.365	0.604	0.442	0.482	0.421	**0.606**
viscoelastic cannula	0.547	0.869	0.976	0.528	0.549	**0.94**
cotton	0.666	0.693	0.952	0.384	0.274	**0.981**
capsulorhexis cystotome	0.563	0.616	0.536	0.449	0.488	**0.541**
Bonn forceps	0.605	0.502	0.643	0.408	0.432	**0.619**
capsulorhexis forceps	0.541	0.844	0.669	**0.53**	0.484	0.42
Troutman forceps	0.974	0.593	0.725	0.453	0.504	**0.542**
implant injector	0.976	0.85	1	0.596	**0.661**	0.508
primary incision knife	0.667	0.476	0.866	**0.687**	0.643	0.588
secondary incision knife	0.821	0.653	0.179	0.583	0.565	**0.669**
micromanipulator	0.43	0.534	0.566	0.5	0.45	**0.592**
Average (mA_z)	0.67	0.647	0.712	0.507	0.528	**0.643**
Standard deviation	0.204	0.138	0.257	0.0849	0.148	0.166

Table 5.5: P-CNN results in terms of areas under the ROC curve (A_z) for the RNT validation subset and the RW testing subset. For each tool, the highest score is marked in bold for the RW data and is underlined for synthetic data.

tool-specific patterns.
Despite the fairly good A_z for the primary incision knife, the confusion matrix shows that this tool is mostly detected as absent. For capsulorhexis cystotome and micromanipulator, the misclassification rates are significant, thus the poor performance. These models were not able to find discriminative features for these tools.

To assess the usefulness of simulated data, we compare the models that are trained on synthetic data and RW data. Thus, we applied the P-CNN approach on the RW data using the same settings used for I-CNN approach (see section 5.4.1). For full details about the application of P-CNN on RW data, we report the results in Appendix B in Table 9.1 along with the confusion matrix in Fig.9.7 and the salient pixels that contribute in the image-level predictions of three surgical tray examples in Fig.9.4, Fig.9.5 and Fig.9.6. We present in Table 5.7 the results of applying the I-CNN and P-CNN on the RW testing subset trained on RW and simulated data. Expectedly, models with deficient performance on the simulated data are badly performing the task on the RW dataset. However, the performance of the cotton has significantly increased compared to the results obtained using models trained on RW data; $A_z = 0.981$ for best performing P-CNN trained on simulated

	RNT validation susbet				RW testing susbet			
Tool	ResNet-152	Inception-V4	Inception-ResNet-V2	NASNet-A	ResNet-152	Inception-V4	Inception-ResNet-V2	NASNet-A
biomarker	0.937	0.871	<u>0.998</u>	0.922	0.83	**0.891**	0.758	0.702
hydrodissection cannula	<u>0.95</u>	0.217	0.446	0.227	0.597	0.584	**0.608**	0.465
viscoelastic cannula	0.648	0.8	<u>0.962</u>	0.916	0.564	0.611	0.598	**0.634**
cotton	0.559	0.555	<u>0.901</u>	0.462	0.762	**0.924**	0.662	0.764
capsulorhexis cystotome	0.328	0.794	<u>0.994</u>	0.623	**0.66**	0.507	0.287	0.565
Bonn forceps	<u>0.296</u>	0.001	0.016	0.004	0.573	0.531	0.489	**0.585**
capsulorhexis forceps	<u>0.687</u>	0.558	0.529	0.472	**0.594**	0.407	0.377	0.331
Troutman forceps	0.006	<u>0.757</u>	0.655	0.669	0.502	0.546	**0.608**	0.426
implant injector	<u>0.988</u>	0.706	0.976	0.702	0.578	**0.649**	0.527	0.504
primary incision knife	0.756	0.768	0.825	<u>0.879</u>	**0.698**	0.571	0.567	0.672
secondary incision knife	0.189	0.164	0.09	<u>0.228</u>	**0.802**	0.686	0.722	0.695
micromanipulator	0.126	0.17	<u>0.501</u>	0.334	**0.562**	0.497	0.512	0.541
Average (mA_z)	0.539	0.53	<u>0.657</u>	0.536	**0.643**	0.617	0.559	0.573
Standard deviation	0.342	0.307	0.347	0.299	0.106	0.154	0.134	0.127

Table 5.6: I-CNN results in terms of areas under the ROC curve (A_z) for the RNT validation subset and the RW testing subset. For each tool, the highest score is marked in bold for the RW data and is underlined for synthetic data.

data and $A_z = 0.772$ for best performing I-CNN trained on RW data (see Table 5.7). This is probably due to the balanced distribution of this tool in the simulated learning subset. The other tools are poorly classified in the RW dataset. As for the biomarker and implant injector, where the models perform well on the simulated data, their models are not able to generalize well on the RW data. This is the case for the viscoelastic cannula where the results obtained are marginally inferior to those obtained using P-CNN models trained on RW data (see Table 5.7). In addition, the I-CNN approach trained on RW data is used as reference distribution to compute the p-value. The P-CNN approach trained on RW data has no meaningful effect on the results ($p = 819 \times 10^{-3}$). The approaches trained on simulated data are considered adversely different from the reference approach ($p = 6 \times 10^{-3}$ for I-CNN, $p = 7 \times 10^{-3}$ for I-CNN).

5.5.5 Surgical Tray Challenges Conclusion

Here, we proposed to generate simulated tray videos along with a patch-based CNN (P-CNN) approach in order to alleviate the inherent challenges of the RW tray

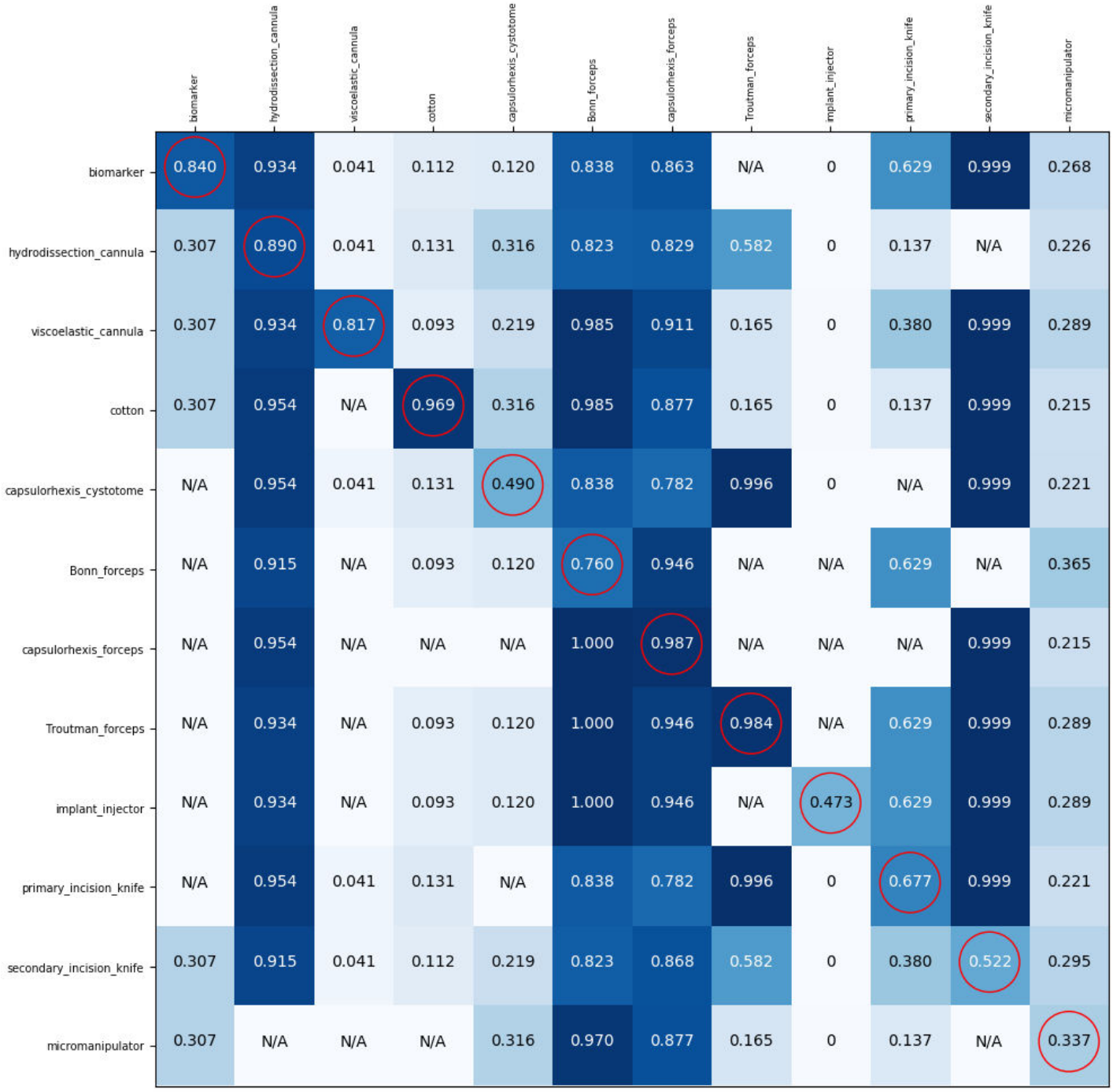


Figure 5.23: Confusion matrix for Inception-ResNet-V2 (P-CNN) tool absence detection of the simulated validation susbet. For easier understanding, the diagonal cells are circled in red. N/A is not applicable: no images were found where the class in row is absent and the class in column is present.

videos: (1) the unbalanced tools distribution. (2) the tools are present concurrently on the tray. (3) the input image sizes impede the CNNs from appropriately performing the task. We recorded simulated tray videos using only the disposable tools in

	Trained on RW data		Trained on simulated data	
Tool	I-CNN	P-CNN	I-CNN	P-CNN
biomarker	0.679	0.763	**0.83**	0.713
hydrodissection cannula	0.827	**0.899**	0.597	0.606
viscoelastic cannula	0.95	**0.973**	0.564	0.94
cotton	0.772	0.381	0.762	**0.981**
capsulorhexis cystotome	0.75	**0.761**	0.66	0.541
Bonn forceps	**0.864**	0.848	0.573	0.619
capsulorhexis forceps	0.768	**0.878**	0.594	0.42
Troutman forceps	0.709	**0.783**	0.502	0.542
implant injector	**0.768**	0.689	0.578	0.508
primary incision knife	0.818	**0.908**	0.698	0.588
secondary incision knife	0.761	**0.896**	0.802	0.669
micromanipulator	**0.856**	**0.856**	0.562	0.592
Average (mA_z)	0.797	**0.802**	0.643	0.643
Standard deviation	0.076	0.154	0.106	0.166
p - value		819×10^{-3}	6×10^{-3}	7×10^{-3}

Table 5.7: I-CNN and P-CNN results of RW testing subset for the best performing networks trained on RW and RNT datasets. For each tool, the highest score is marked in bold.

which we intensively imitated the surgeons actions. Rather than using the whole image, we carried out the classification using patches (P-CNN). Three networks were trained on the simulated data, then evaluated using RW data. The experiments show that the models were not able to perform properly the task on the simulated data, and afterwards on the RW data. Indeed, training on simulated data and evaluating on RW data was efficient for only one tool. This tool underlined the high potential of the simulated datasets in boosting the performance in such complicated dataset. However, some tools models were not able to generalize well on RW data. This can be justified by the complexity of the real world tray scene; it is infeasible to render tray videos for every specific real world use-case. In addition, some other tools models have poorly carried on the task on the simulated data. This is primarily due to the tools co-occurrences in the simulated data, which were easily learnt by the CNNs; the P-CNN approach was not sufficient in preventing the CNNs from looking for conjunctions of tools.

5.6 Summary

In this chapter, we have presented a deep learning solution to address the surgical tool presence detection in the cataract surgery videos. On the tool-tissue interaction videos, four well-known networks were used to train and evaluate the models using the I-CNN approach. The models have efficiently performed the task, notably using NASNet-A. Indeed, we have shown that the features learnt by the networks are

related to the deformation and motions issued from the interaction between the tools and the eyes as well as to the tools themselves. On the surgical tray videos, similar strategies to chapter 4 were applied to acquire the tool signals. For detecting tool presence on the surgical tray, the I-CNN approach yielded moderate performance for most of the tools. This was due to the tools distribution in the dataset, to the tools specification (tools are concurrently present with high similarity in shape for some of them) and to the input image size settings. To address these challenges, we have generated simulated surgical tray scenes and we proposed a patch-based CNN approach. However, P-CNN models were not able to properly perform the task on the simulated data, resulting in poor performance on RW data with the exception of one tool. This tool has demonstrated that the simulated data can be a useful complement to the real world data. However, the manual construction of simulated tray data under the required constraints is not a simple task because some tools models have poorly performed the task due to tools co-occurrences found by the CNNs. Furthermore, the sophistication and the variability of the real world tray data are another impeding factor for reaping the benefits of the artificial data in the surgical tool presence detection task. On the other side, the I-CNN showed very good performance in detecting the tools changes. It was subject to some limitations by reason of the surgeons hands.

The high performance on the tool-tissue interaction videos have a high potential to be used in real world applications. Albeit these satisfactory results, the surgical tray tools information are still worthwhile because with very good performance on the tray, such as the change detection results, we are predominantly capable of improving the tools presence accuracy in the surgical field by enforcing the temporal constraints of the surgical workflow on both video types.

"Continuous improvement is better than delayed perfection."

Mark Twain

6

Temporal Analysis of Surgery Videos

Chapter Content

Automatic video analysis has gained a lot of attention with the emergence of deep learning. Different strategies have been proposed for this task (see section 2.2). One of the most efficient strategies is to combine a CNN with a RNN. In other words, the CNN analyses the 2-D images and the RNN analyses the temporal information. A variant of RNN assures the ability of taking of advantage of long-term relationships between events. In chapter 5, we studied the ability of using only the visual features in order to perform surgical tool presence detection. In addition to the visual features, the surgical workflow enforces some temporal constraints which can be used to improve surgical tool detection performance. In this chapter, we present an extension to the pipeline, discussed in chapter 5, that reaps the benefits of the temporal information in order to boost the performance of the surgical tool detection system. This can be done by adding a RNN on top of the CNN. Various

strategies are applicable in this thesis. These strategies consist of analysing the temporal information of: (1) the tool presence in the tool-tissue interaction videos. (2) the tool presence in the surgical tray videos. (3) the tool presence in both video types jointly.

In section 6.1, we present a comprehensive review on sequence classification using neural networks. The section 6.2 discusses the different strategies studied in order to take advantage of the temporal information in both video types.

6.1 Sequence Classification with Neural Networks

The vanilla neural network or the CNN consider the input vectors independent of each other. However, in many applications, the input or the output space is a sequence of vectors. Conceptually, the ANNs and the CNNs are inadequate for tackling any sequential data problem. In this section, we discuss a special kind of neural network architectures designed to process sequential data: Recurrent Neural Network (RNN), and its most common updated version Long Short Term Memory (LSTM).

6.1.1 Recurrent Neural Network

Recurrent neural network is a neural network that have feedback loops, making it capable of processing a sequence of vectors $x_1, ..., x_T$. RNN can receive one value or a sequence of values as input, and it can also produce one value or a sequence of values as output. The one-to-one (one value as input to one value as output) RNN is the vanilla neural network. In addition, there exist one-to-many and many-to-one RNN approaches. They have shown their efficiency in many applications, such as image captioning and sentiment classification [Baktha and Tripathy, 2017] [Johnson et al., 2016]. In this thesis, we are interested in the many-to-many RNN approach, which is illustrated in Fig.6.1. The RNN has an internal hidden state h which is updated every time the RNN reads a new input. This hidden state is fed back to the model the next time it reads an input. This hidden state can be deemed as a "memory" capturing information concerning all previous time steps in a sequence. The hidden state can be formulated as:

$$h_t = f\left(W\begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} + B\right) \tag{6.1}$$

where h_{t-1} is the hidden state in the previous RNN layer and x_t is the input vector at time step t. The first hidden state h_0 is typically initialized to all zeros, but it can be treated as parameters to be learnt as well. h_{t-1} and x_t are concatenated and transformed linearly by the parameters W and B, then squashed by the non-linearity f. The most common non-linear function used in RNN is $tanh$. W is the concatenation of the two matrices W_h and W_x (see Fig.6.1). Thus, the Equation (6.2) can be expressed as:

$$h_t = tanh(W_h h_{t-1} + W_x x_t + B) \tag{6.2}$$

It is worth noting that the same parameters are used at every time step, thus the employment of the same f for every time step in Fig.6.1. For timestamp $t = 1...T$, y_t is considered the output layer of x_t as in the ANNs.

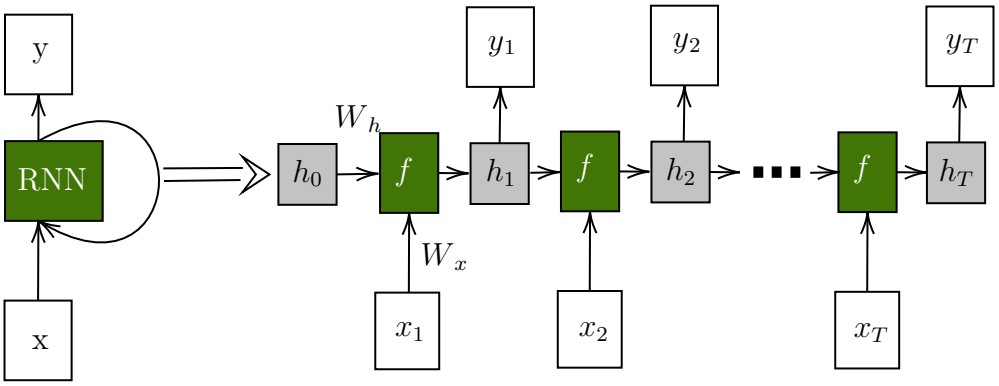


Figure 6.1: Illustration of a many-to-many recurrent neural network, where the input and the output are a sequence of vectors. Green boxes represent the hidden states that manipulates a set of internal variables h_t based on previous hidden state h_{t-1} and the current input using the Equation (6.2).

In theory, RNNs are able to handle long term dependencies, i.e. long sequences of data. Nonetheless, RNNs are practically not able to handle them. This problem was explored by [Bengio et al., 1994], which indicates that the fundamental reason behind this failure is the vanishing gradient problem. This issue leads to exponentially small gradients and a decay of information through time steps.

6.1.2 Long Short Term Memory

To address the vanishing gradient problem, the solution proposed in [Hochreiter and Schmidhuber, 1997] is to use gating. Gating is a technique that helps the network decide when to forget the current input, and when to remember it for future time steps. Using this principle, the Long Short Term Memory (LSTM) is designed to remember information for long periods. In Fig.6.2, we present the difference between the internal modules of RNN and LSTM. In addition to the hidden state h_t, LSTM has also a cell state c_t. In each time step, the LSTM has the ability to add or remove information from the cell state using gating mechanisms. They are represented by a sigmoid layer and a pointwise multiplication. This mechanism can be expressed as:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \tag{6.3}$$

leading to:

$$c_t = f \odot c_{t-1} + i \odot g \tag{6.4}$$

$$h_t = o \odot tanh(c_t) \tag{6.5}$$

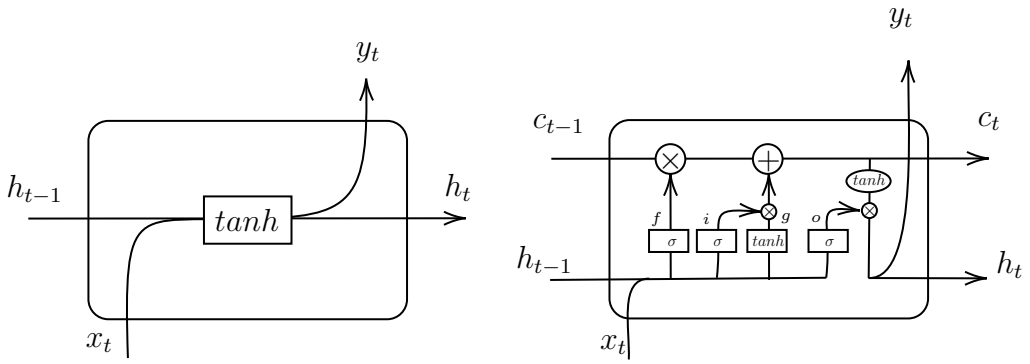


Figure 6.2: **Left**: the structure of the module in RNN. **Right**: the structure of the module in LSTM.

The vectors i, f, o are thought as binary gates where i is used to control whether a memory cell is updated, f is for controlling whether it is reset to zero and o the local state of the cell is revealed in h. These three gate functions allow the gradient on the memory cell c to flow backwards sustainably for a long sequences of data.

6.1.3 Bidirectional Recurrent Neural Network

In RNNs, the hidden state h_t is expressed in terms of the previous states. However, for many applications, the future states contain discriminative information, thus the need of bidirectional recurrent neural network (BRNN) [Schuster and Paliwal, 1997]. In BRNN, the future information is incorporated with the previous states to evaluate the current state. In practice, BRNN is just putting two independent RNNs together. This structure allows the BRNN to have both backward and forward information at every time step by simply connecting two hidden layers of opposite directions to the same output. The same concept is applicable as well for LSTM cells, as illustrated in Fig.6.3.

6.2 Temporal Analysis for Tool Presence Detection

The temporal information is a key component for any computer-assisted surgical system, and subsequently for the surgical tool presence detection task. Indeed, the surgical tools are often used in a predefined order. Exploring the temporal dependencies is closely pertinent for differentiating them, in particular the tools that are similar to one another (e.g. the knifes, the cannulae or the forceps). Therefore, it seems particularly useful to guide CNN training based on the temporal context. In addition, taking the temporal sequencing into account is worthwhile: knowing which tools have already been used since the beginning of the surgery considerably helps in recognizing which tools are currently being used.

In chapter 5, the CNNs results were obtained using only the visual features existed in the video frames. However, this section is dedicated to model the temporal

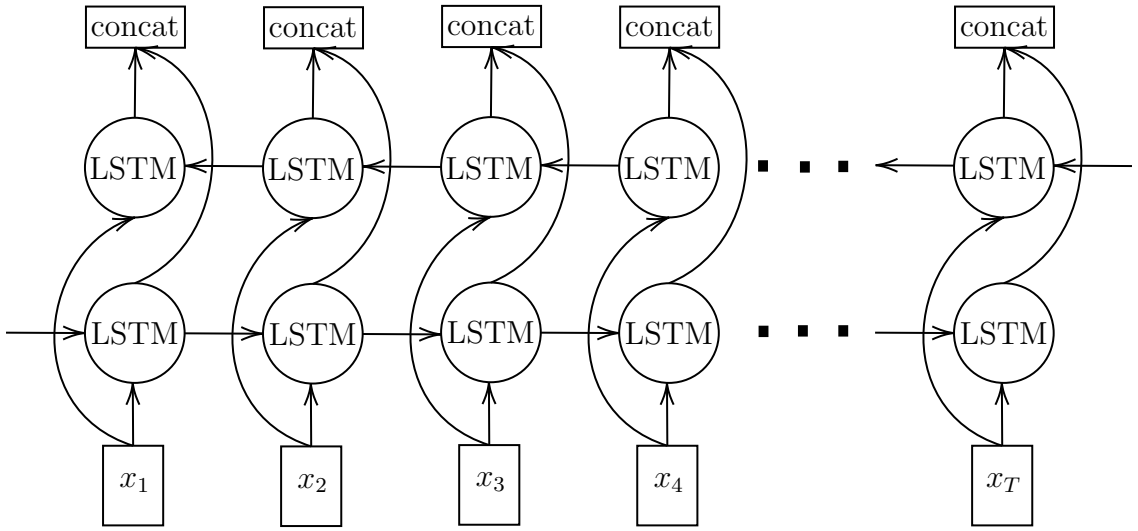


Figure 6.3: structure of BRNN using LSTM cells.

dependencies among frames by reinforcing the CNN results using RNN to improve the recognition accuracy in the surgical field. In other words, we study the feasibility of using the temporal information to extend the pipeline from only CNN to "CNN+RNN" for the task of automatic tool usage annotation. This implies harnessing the visual and temporal information separately. In this section, we refer to the best CNN results of tool presence detection on the tool-tissue interaction videos (i.e. in the microscope field of view) as "MicroTP", of tool presence detection on the surgical tray videos as "TrayTP" and of the tools changes detection on the surgical tray videos as "TrayCD".

Four different approaches can be applied to improve the results of the previous chapter. One is to enforce the temporal constraints on the tool-tissue interaction videos, referred as "T(MicroTP)". In addition, we study two approaches of fusing information across temporal domain: the fusion can be done using the tools presence signals obtained on both video types, referred as "T(MicroTP+TrayTP)", or it can be done by fusing the tools presence signals obtained for the tool tool-tissue interaction videos with the tools change detection predictions, referred as "T(MicroTP+TrayCD)". Despite the main interest of improving the accuracy of surgical tool presence detection in the tool-tissue interaction videos using "MicroTP", "MicroTP+TrayTP" and "MicroTP+TrayCD", it is interesting to explore the temporal information on the surgical tray as well, which is referred as "T(TrayTP)".

6.2.1 Models Formulation

In a typical "CNN+RNN" approach, the RNN processes the visual features extracted from the video frames by the CNN. In order to reduce the complexity, we propose in this study to use the output predictions of the CNNs as input to the RNN to analyze the temporal sequencing for the entire surgery.
Let I_t denote the t-th frame in a video of the training data. Suppose there are r tools

to be classified, the CNN predictions of I_t can be expressed as $p(I_t) = \{p_1, ..., p_r\}$. The input sequence of the RNN is $p(I_t)$, the predictions of the CNNs for each frame in a video. The output vector computed by the RNN is denoted as $o(I_t) = \{o_1, ..., o_r\}$. The network is structured in such a way that the output vector $o(I_t)$ depends on the previous output vectors $o(I_v), v < t$ as well as to the input vector $p(I_t)$. In a one-layer RNN, each input element $p(I_t)$ is connected to a group of neurons C_t called "cell". Those neurons are connected to the output elements $o(I_t)$ as well as to the neurons of the next cell C_{t+1}. They share the same weights during training. In a multi-layer RNN, each timestamp t is associated with multiple cells $C_{(i,t)}$, where $i \in \{1, ..., n\}$ is the layer index. At each timestamp t, $p(I_t)$ is connected to $C_{(1,t)}$, $C_{(i,t)}$ is connected to $C_{(i+1,t)}$ for $i = 1..n-1$, and $C_{(n,t)}$ is connected to $o(I_t)$. In each layer i, $C_{(i,t)}$ is connected to $C_{(i,t+1)}$. Weights are shared across all cells in the same layer. For a bidirectional RNN, two independent RNNs follow the same process described previously with only one difference: information flows from timestamp t to timestamp $t+1$ in one of them; information flows from timestamp t to timestamp $t-1$ in the other one. Their outputs are concatenated and connected to the $o(I_t)$.

This process is used in the "T(MicroTP)" and "T(TrayTP)" approaches. The input sequence is different for "T(MicroTP+TrayTP)" and "T(MicroTP +TrayCD)" approaches, however, the output vector of the RNN is always $o(I_t)$. In regards to "T(MicroTP+TrayTP)", the CNN predictions of I_t for both video types are fused at each timestamp t, resulting in $2 \times r$ elements in the input vector. As for "T(MicroTP+TrayCD)", the tools changes signals for an action (2 images) at second s are converted to tool appearances and disappearances signals by using the hypothesis followed in section 5.3.1: the first image of the actions is containing the tools disappeared and the second image of the action is containing the tools appeared. Then, this information is fused with $p(I_t), s-1 < t \leq s$, resulting in a input sequence of $r + 2$ elements.

6.2.2 Experimental Setups

In this section, we present the settings used in order to exploit the temporal information in the surgical tool detection system. First, we describe the datasets used to perform the task. Afterwards, we present the RNN configurations applied to this task.

6.2.2.1 Datasets

For "T(MicroTP)" and "T(MicroTP+TrayCD)", the RW dataset was divided similarly to the CNN pipeline (see sections 5.3.2.1 and 5.4.1.1). Since the results are quite good on the training and testing subsets for tool presence detection on the tool-tissue interaction videos and for change detection on the surgical tray videos, it is feasible to perform the training on the same training videos used in the CNN pipeline: 23 videos for learning, 2 videos for validation and 25 videos for testing. Nevertheless, the results of the tool presence detection on the tray videos were very well on the training subset and poor on the testing subset (overfitting problem). Thus, only the testing subset (25 videos) of the CNN pipeline is used to train and

test the "T(TrayTP)" and "T(MicroTP+TrayTP)" approaches: 23 videos for learning and 2 videos for validation.

6.2.2.2 Network Configurations

The RW cataract surgical videos contain long sequences since each video records the entire surgery. Training long-term relationships with RNNs is computationally expensive using long sequences. With these considerations, instead of using the complete video as input sequence to the RNN, we propose to subsample in time the surgical videos into short video clips, thus analyzing shorter sequences using the RNNs. In that purpose, M subsampled versions of each original sequence V, denoted by $V^{(m)}, m = 1..M$, are generated as follows:

$$V^{(m)} = \{V_u | u = m + t \quad M, t \in N^*, u \leq |V|\} \tag{6.6}$$

During the training and the testing phases, each of the M subsequences of V are analyzed independently whereas the final prediction sequence for V during testing is obtained by interleaving the resulting M prediction sequences. This sort of data augmentation is applied for both video types except with the approach "T(MicroTP+TrayCD)" where it is only applied on the tool-tissue interaction videos.

For all approaches, the CNN outputs of each frame are fed to BRNN in order to take temporal relationships between events into account. The cross-entropy function, detailed in Equation (5.6), is used to compute the loss, since it is regarded as a multi-label classification task. For "T(TrayTP)" and "T(MicroTP+TrayTP)", we perform the task using leave-two-out cross-validation. We used a one-layer RNN with LSTM cells. The number of neurons per cell is 128. The RNNs were trained using the RMSProp algorithm with a constant learning rate of 0.001. RNNs were implemented using Keras version 2.0.8.

6.2.3 Experimental Results

Here, the BRNN is used to get information from past and future states. We present the tool presence detection results of "CNN+RNN" in offline mode in Table 6.1 and Table 6.2 on the tool-tissue interaction videos and the surgical tray videos, respectively. In this study, we use the CNN outputs of the best performing networks: NASNet-A (I-CNN) for the tool-tissue interaction videos, ResNet-152 (I-CNN) for the tools change detection on the tray and Inception-ResNet-152 (P-CNN) for tool presence detection on the surgical tray. In addition, the MicroTP and TrayTP approaches are used as reference distribution to compute the p-value.

On the tray, the "CNN+RNN" has significantly improved the tool detection results; $A_z = 0.74$ for "TrayTP" whereas the performance of "T(TrayTP)" is $A_z = 0.825$. It is worth mentioning that three tools are only present in one video of the tray testing subset videos: they are only present in either the learning subset or the validation subset during the training phase. For these tools, the "CNN+RNN" is not applicable in the "T(TrayTP)" and "T(MicroTP+TrayTP)" approaches. In addition, one might notice that the "CNN+RNN" has moderately improved the

CNN results on the tool-tissue interaction videos (see Table 6.1). Interestingly, the temporal information is slightly improving the tool presence detection in the surgical field; $A_z = 0.983$ for "MicroTP" whereas the performance of "T(MicroTP)" is $A_z = 989$. The employment of the temporal information has scarcely affected the results because the CNN results are already very good.

Tool	MicroTP	T(MicroTP)	T(MicroTP+TrayTP)	T(MicroTP+TrayCD)
biomarker	0.954	0.98	0.68	**0.989**
Charleux canula	0.96	**0.975**	0.961	**0.975**
hydrodissection canula	0.98	0.983	0.965	**0.986**
Rycroft canula	0.989	0.993	**0.996**	0.994
viscoelastic cannula	0.962	0.97	**0.979**	0.974
cotton	0.991	0.993	0.898	**0.995**
capsulorhexis cystotome	0.998	**0.999**	**0.999**	**0.999**
Bonn forceps	0.98	**0.993**	0.991	0.992
capsulorhexis forceps	0.987	0.996	0.971	**0.997**
Troutman forceps	0.988	**0.995**	0.99	0.994
needle holder	0.991	0.989	**0.994**	0.988
irrigation/aspiration handpiece	0.996	0.997	**0.998**	0.997
phacoemulsifier handpiece	0.998	**0.999**	N/A	**0.999**
vitrectomy handpiece	0.957	0.96	N/A	**0.962**
implant injector	0.976	0.989	**0.994**	0.993
primary incision knife	0.981	**0.996**	0.995	0.995
secondary incision knife	0.997	**0.999**	**0.999**	**0.999**
micromanipulator	0.995	**0.997**	0.994	0.996
suture needle	0.975	**0.998**	0.998	0.997
Mendez ring	0.991	0.984	N/A	**0.995**
Vannas scissors	0.984	**0.984**	0.962	0.964
Average (mA_z)	0.983	0.989	0.965	**0.99**
Standard deviation	0.01	0.01	0.075	0.011
p - value		1.5×10^{-3}	288.4×10^{-3}	9.1×10^{-3}

Table 6.1: "CNN+RNN" results in terms of A_z for "MicroTP", "MicroTP+TrayTP" and "MicroTP+TrayCD" approaches. For each tool, the highest score is in bold. N/A is not applicable: the tools are only present in one video of the tray dataset.

In regards to the information fusion approaches, the poor results of the tool presence on the tray (m$A_z = 0.74$) are adversely affecting the very good results

obtained in the operative field (mA_z = 0.983), resulting in mA_z = 0.965 for "T(MicroTP+TrayTP)". However, the results show good improvements for some tools that are very well detected on the tray, for instance the viscoelastic canula with A_z = 0.979 which is the highest score among the different approaches. In "T(MicroTP+TrayCD)", the incorporation of tool presence signals in the tool-tissue interaction videos and the tools changes signals on the tray over temporal dimension has greatly performed the task with mA_z = 0.99. This result is slightly better than exploring the temporal information on only the tool-tissue interaction videos. These results demonstrate that the tray information is worthwhile; they are able to improve the tool presence detection performance in the surgical field.

Tool	TrayTP	T(TrayTP)
biomarker	**0.763**	0.694
Charleux canula	0.415	**0.648**
hydrodissection canula	**0.899**	0.884
Rycroft canula	**0.832**	0.812
viscoelastic cannula	**0.973**	0.954
cotton	**0.381**	0.359
capsulorhexis cystotome	0.761	**0.831**
Bonn forceps	0.848	**0.928**
capsulorhexis forceps	**0.878**	0.7
Troutman forceps	**0.783**	0.773
needle holder	0.576	**0.997**
irrigation/aspiration handpiece	0.871	**0.929**
implant injector	0.689	**0.793**
primary incision knife	**0.908**	0.868
secondary incision knife	**0.896**	0.817
micromanipulator	0.856	**0.975**
suture needle	0.488	**0.895**
Vannas scissors	0.512	**0.992**
Average (mA_z)	0.74	**0.825**
Standard deviation	0.185	0.156
p - value		7×10^{-2}

Table 6.2: "CNN+RNN" results on the tray videos. For each tool, the highest score is in bold.

6.2.4 Temporal Analysis Conclusion

In this study, we have proposed an extension to the pipeline proposed in chapter 5 by exploring the temporal constraints of the surgical videos. The objective is to improve the CNN results in the tool presence detection task. Two straightforward approaches ,"T(TrayTP)" and "T(MicroTP)", are proposed to improve the results on the tool-tissue interaction videos and the surgical tray videos separately. For information fusion, we proposed another two approaches, "T(MicroTP+TrayTP)"

and "T(MicroTP+TrayCD)", to improve the results in the surgical field. These approaches are all based on applying a BRNN on the outputs of the CNNs. Compared to the CNN results, the "T(TrayTP)" approach has improved the tool presence results on the tray with $mA_z = 0.825$, which implicitly points out to the significance of the temporal information. However, the "T(MicroTP)" has marginally refined the performance on the tool-tissue interaction videos because the CNNs have already performed efficiently the task. Despite the very good results of MicroTP, "T(MicroTP+TrayCD)" has yielded slightly better results, however, "T(MicroTP +TrayTP)" has yielded inferior results because the TrayTP has poorly performed the task. These experimentations have demonstrated that the surgical tray contains worthy information and with greater performance on the tray, we can predominately improve more the tool presence results in the operative field ($p = 9.1 \times 10^{-3}$).

On the down side, we have only presented the results in offline mode using the bidirectional RNN which takes advantage of past and future information. However, in Appendix C, we report the results in online mode but under completely different configurations.

6.3 EndoVis/CATARACTS Subchallenge

With 14 participating teams, the first edition of CATARACTS in 2017 (see section 3.4.1) was considered a success. The top ranking solutions achieved very good tool recognition performance. However, human annotators still outperform the automatic solutions. Therefore, we decided to repeat this experience by adding a new technical challenges. The tool-tissue interaction videos (already released in CATARACTS 2017) along with the surgical tray videos are released publicly in the context of a sub-challenge in the International Conference On Medical Image Computing Computer Assisted Intervention (MICCAI). This second edition of CATARACTS[1] is organized as a sub-challenge of the MICCAI 2018 EndoVis challenge. This new technical challenge can assist in pushing further the performance in the surgical field, possibly outperforming the human annotators, along with an interesting methodological challenge: information fusion. We have roughly 30 registered users up to this point.

[1] `https://cataracts2018.grand-challenge.org/`

"One... moment you think it's the end. But it isn't, it's just the beginning."

Deyth Banger

7 Discussions and conclusions

Chapter Content

During the last decade, we have witnessed rapid growth in the size of medical data archives, however, they are relatively unexplored nowadays. This data is considered an essential factor for any computer-assisted surgery system. In this thesis, we are interested in reusing these archives in order to automate the process of extracting information from the medical data. This can be deemed a stepping stone towards the computer-assisted surgery systems. Particularly, we are interested in the activity recognition task in the operating rooms (ORs). Here, we focus on using visual information issued from cameras. Indeed, if we are able to recognize the surgeon's activities at each instant of the surgery, we can automatically determine what kind of help the surgeons need, if any. This can be done by analysing the surgical videos recorded during the surgeries in order to generate alerts and recommendations. Recent studies have proved that the tool usage signals perform better than the visual features in the activity recognition task. However, acquiring the tool usage information is still a challenging task during surgeries. That is, the objective of this thesis is to automatically recognize the tools in the surgical videos.

7.1 Summary and Discussions

In this dissertation, we focused on the cataract surgery because it is the most common eye surgery around the world. However, numerous challenges are present in the surgical field: the tools are partially visible and they resemble strongly. Therefore,

we have proposed the addition of a second video stream filming the surgical tray. In chapter 3, we presented a dataset of 50 real-world cataract surgeries, each of which is recorded in two videos: tool-tissue interaction videos and surgical tray videos. The former captures the anatomical structure of the patient's eye, while the latter captures the task done by surgeons on the surgical tray (i.e. take a tool from the tray or put a tool on the tray, etc.). As for the ground truth, we have built a web-application in order to annotate the tool usage signals for both video types. This dataset has allowed us to evaluate our approaches in the surgical tool recognition task.

In chapter 4, we introduced two different tasks to address the surgical tool detection on the surgical tray videos. The first one is to detect the tools changes: the tools put on and taken from the surgical tray. The second task is the tool presence detection at each instant of the surgery. A similar patch-based pipeline was proposed for both tasks. This pipeline consists of: (1) extracting handcrafted or shallow learning features and (2) pixel-wise classification using k-NN regression. Due to the complexity of acquiring the ground truth in this case, we evaluated the models using a small subset of images extracted from the RW dataset. For the tools changes task, we have shown that the shallow learning features yield promising results with $A_z = 0.959$. However, for the tool presence detection task, the features extracted were not as discriminative as required, thus the performance in this case was abysmal with $\mathrm{m}A_z = 0.6$.

In chapter 5, we proposed to use deep learning for surgical tool presence detection task on both video types. This eliminates the need of manual features engineering. We have presented the best network architectures to date for ImageNet classification challenge, which were used to perform the tool presence task on both video types. A default configuration of these CNNs (I-CNN) was first used to address this problem. For the tool-tissue interaction videos, the networks have perfectly carried out the task. In addition, the visual features extracted by the networks were not only related to the tools but also to the way they interact with the eye. As for the surgical tool detection on the tray videos, the networks have poorly performed the task due to several reasons: (1) the unbalanced tools distribution which had a severely negative impact on overall performance. (2) most of the tools are present concurrently on tray. (3) the input image size settings used in I-CNN is an impeding factor to differentiate the tools. In this light, we have proposed to generate simulated surgical tray scenes along with a patch-based CNN to alleviate these challenges. Both propositions were not widely able to overcome the inherent challenges in the surgical tray videos. The models have learnt sometimes the tools co-occurrences and they were not able to generalize well on RW data. There were some exceptions (e.g. cotton) that have proved the complementarity nature of the synthetic data regarding the real-world data. Additionally, the P-CNN has marginally improved the results, however, it was not efficiently able to address all the problems. Quantitative results comparing the best performing CNN model with the patch-based pipeline proposed in chapter 4 are shown in Table 7.1. One can obviously notice that the deep learning pipeline was more effective than the patch-based pipeline, and subsequently the learning features were more discriminative than the shallow learning features. In regards to the tools changes detection, we proposed a deep learning pipeline which have yielded very

Tool	Patch-based pipeline	Inception-ResNet-152 (P-CNN)
hydrodissection canula	0.659	**0.899**
viscoelastic canula	0.491	**0.973**
cotton	**0.961**	0.381
capsulorhexis cystotome	0.606	**0.761**
Bonn forceps	0.552	**0.848**
capsulorhexis forceps	0.354	**0.878**
Troutman forceps	0.485	**0.783**
implant injector	0.619	**0.689**
primary incision knife	0.663	**0.908**
secondary incision knife	0.586	**0.896**
micromanipulator	0.618	**0.856**
Average mA_z	0.6	**0.807**
p-value		5.5×10^{-3}

Table 7.1: Comparison between the best performing CNN results and the patch-based pipeline results (chapter 4) for surgical tool presence detection on the surgical tray. For each tool, the highest score is in bold.

good performance with $A_z = 0.956$. Compared to the results obtained in chapter 4, the CNN-based solution yielded inferior results than the block-matching approach ($A_z = 0.959$). However, these approaches were trained differently: images containing many changes (i.e. the time period is up to tens of seconds) for the patch-based approach and images containing few changes (i.e. the time period is one second) for the CNN-based solution. Technically speaking, these approaches can not be compared due to the difference in the training images. In terms of computations, the deep learning solution is much faster than the patch-based approach. The CNN-based solution processes up to tens of images per second whereas the patch-based approach processes one image in at least one second. With these considerations, this deep learning solution is generally more feasible in this case.

In Chapter 6, we have proposed an extension to the CNN models proposed in chapter 5 in order to include the temporal information ("CNN+RNN"). Four different approaches were proposed to explore the temporal constraints. In accordance with the aim of the thesis, three of them were proposed to improve the surgical tool presence detection in the surgical field. The last one was proposed to analyse the

temporal information on the surgical tray. The experiments highlighted the significance of the temporal information in this context. It has considerably boosted the performance on the surgical tray videos. The temporal analysis in the surgical field has slightly improved the results, however, the best results was obtained with the information fusion between the tool usage signals in the operative field and the tools changes signals on the tray. This has underlined the significance of the surgical tray information in this context.

7.2 Conclusions and Future Works

Similar to numerous studies, this thesis has demonstrated once again that deep learning is considerably more effective than shallow learning features, thus its domination on computer vision domain. Notwithstanding the satisfactory results in the microscope field of view, the surgical tray is still considered worthwhile. The information fusion of both video streams have demonstrated the utility of the surgical tray despite its slight improvement. It was obtained using solely the tools changes signals, which can be considered a scarce amount of information compared to the actual amount of information on the tray. Future work should address this limitation, so instead of detecting the tools changes, recognizing the tools changes (i.e. recognizing the tools put on or taken from the surgical tray) would reap maximum benefits of the surgical tray information. In addition, future work should address the challenges of the surgical tool presence detection on the tray. The active learning would help in annotating the tools at pixel-level in the images, which would reduce the region of interests for the CNNs. It would produce better discriminative features, thus better performance.

Two different challenges, issued from this work, were proposed to the community. CATARACTS 2017 has drawn the attention of the community, in which we received 14 different solutions. This work has spurred the community to the research in the surgical tool detection domain, i.e. papers addressing this problem are going to be published shortly. In addition, we have proposed a pipeline based on deep learning to address the surgical tool presence detection problem in cataract surgery videos. In accordance with the thesis objective, the proposed system is competently performing the task. Only the offline version of the system is presented. Numerous applications can be envisaged using this work. It would be helpful for report generation and surgical workflow optimization by evaluating the quality of the surgical procedure performed. This implicitly means recognizing the steps followed by the surgeons along the surgery. Another interesting application is the surgical skill assessment. Using the analysis of the surgical procedure, we can possibly evaluate the skills of the young surgeons. We can quantify their performance while providing appropriate advices. This work can also be used in automatic indexing of medical videos. This application facilitates the search for a specific task within the surgical videos. However, the online version, which have expectedly yielded slightly inferior results than the offline version (see Appendix C), is the solution to generate warnings and recommendations along the surgery. This would help with the decision-making process during the surgery, especially for young surgeons. Using these encouraging results, automatic surgery monitoring system [Charrière et al., 2017] is now feasi-

ble. Indeed, this work can be deemed as a step forward to the implementation of such system. For instance, it would be interesting to automatically recognize the abnormal events or the events that may lead to abnormal situations. Using the actions performed by the experienced clinicians in similar situations, we can define recommendations, trigger alerts and propose actions, which is really helpful for the education of young surgeons. Constructive discussions with clinicians about this point can produce a feasible clinical application. Additionally, this work can be integrated in surgical gesture simulators, to provide guidance to surgeons who are training on specific gestures.

8
Publications

Hassan Al Hajj, Mathieu Lamard, Béatrice Cochener, Gwenolé Quellec. Surgical tool detection for cataract surgery monitoring. Abstract & poster à IEEE EMBC 2016.

Hassan Al Hajj, Mathieu Lamard, Béatrice Cochener, Gwenolé Quellec. Smart data augmentation for surgical tool detection on the surgical tray. Proc. IEEE EMBC. 2017 Jul;4407-10.

Hassan Al Hajj, Mathieu Lamard, Katia Charrière, Béatrice Cochener, Gwenolé Quellec. Surgical tool detection in cataract surgery videos through multi-image fusion inside a convolutional neural network. Proc. IEEE EMBC. 2017 Jul;2002-5. (see Appendix C)

Hassan Al Hajj, Mathieu Lamard, Pierre-Henri Conze, Béatrice Cochener, Gwenolé Quellec. Monitoring Tool Usage in Surgery Videos using Boosted Convolutional and Recurrent Neural Networks. Medical Image Analysis, July 2018, 27: 203-218. (see Appendix C)

Hassan Al Hajj, Mathieu Lamard, Pierre-Henri Conze, Soumali Roychowdhury, Xiaowei Hu, Gabija Maršalkaitė, Odysseas Zisimopoulos, Muneer Ahmad Dedmari, Fenqiang Zhao, Jonas Prellberg, Manish Sahu, Adrian Galdran, Teresa Araújo, Duc My Vo, Chandan Panda, Navdeep Dahiya, Satoshi Kondo, Zhengbing Bian, Arash Vahdat, Jonas Bialopetravičius, Evangello Flouty, Chenhui Qiu, Sabrina Dill, Anirban Mukhopadhyay, Pedro Costa, Guilherme Aresta, Senthil Ramamurthy, Sang-Woong Lee, Aurélio Campilho, Stefan Zachow, Shunren Xia, Sailesh Conjeti, Danail Stoyanov, Jogundas Armaitis, Pheng-Ann Heng, William G. Macready, Béatrice Cochener, Gwenolé Quellec. CATARACTS: Challenge on Automatic Tool Annotation for cataRACT Surgery **(submitted to Medical Image Analysis)**.

(see Appendix C)

Bibliography

[Abadi et al., 2016] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv:1603.04467 [cs]*.

[Abdi and Williams, 2010] Abdi, H. and Williams, L. J. (2010). Principal component analysis. *WIREs Comp Stat*, 2(4):433–459.

[Agarwal et al., 2005] Agarwal, A., Jawahar, C. V., and Narayanan, P. J. (2005). A Survey of. Technical report. Centre for Visual Information Technology.

[Ahmadi et al., 2006] Ahmadi, S.-A., Sielhorst, T., Stauder, R., Horn, M., Feussner, H., and Navab, N. (2006). Recovery of surgical workflow without explicit models. *Med Image Comput Comput Assist Interv*, 9(Pt 1):420–428.

[Allan et al., 2013] Allan, M., Ourselin, S., Thompson, S., Hawkes, D. J., Kelly, J., and Stoyanov, D. (2013). Toward Detection and Localization of Instruments in Minimally Invasive Surgery. *IEEE Transactions on Biomedical Engineering*, 60(4):1050–1058.

[Allan et al., 2014] Allan, M., Thompson, S., Clarkson, M. J., Ourselin, S., Hawkes, D. J., Kelly, J., and Stoyanov, D. (2014). 2d-3d Pose Tracking of Rigid Instruments in Minimally Invasive Surgery. In *Information Processing in Computer-Assisted Interventions*, Lecture Notes in Computer Science, pages 1–10. Springer, Cham.

[Amin et al., 2016] Amin, J., Sharif, M., and Yasmin, M. (2016). A Review on Recent Developments for Detection of Diabetic Retinopathy. *Scientifica (Cairo)*, 2016:6838976.

[Baktha and Tripathy, 2017] Baktha, K. and Tripathy, B. K. (2017). Investigation of recurrent neural networks in the field of sentiment analysis. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 2047–2050.

[Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer.

[Bengio et al., 1994] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Netw*, 5(2):157–166.

[Bharathan et al., 2013] Bharathan, R., Aggarwal, R., and Darzi, A. (2013). Operating room of the future. *Best Pract Res Clin Obstet Gynaecol*, 27(3):311–322.

[Bhatia et al., 2007] Bhatia, B., Oates, T., Xiao, Y., and Hu, P. (2007). Real-time Identification of Operating Room State from Video. In *Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence - Volume 2*, IAAI'07, pages 1761–1766, Vancouver, British Columbia, Canada. AAAI Press.

[Bilinski et al., 2013] Bilinski, P., Corvee, E., Bak, S., and Bremond, F. (2013). Relative dense tracklets for human action recognition. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7.

[Blum et al., 2010] Blum, T., Feussner, H., and Navab, N. (2010). Modeling and segmentation of surgical workflow from laparoscopic video. *Med Image Comput Comput Assist Interv*, 13(Pt 3):400–407.

[Blum et al., 2008] Blum, T., Padoy, N., Feußner, H., and Navab, N. (2008). Workflow mining for visualization and analysis of surgeries. *Int J CARS*, 3(5):379–386.

[Bodenstedt et al., 2017] Bodenstedt, S., Wagner, M., Katić, D., Mietkowski, P., Mayer, B., Kenngott, H., Müller-Stich, B., Dillmann, R., and Speidel, S. (2017). Unsupervised temporal context learning using convolutional neural networks for laparoscopic workflow analysis. *arXiv:1702.03684 [cs]*.

[Bouget et al., 2017] Bouget, D., Allan, M., Stoyanov, D., and Jannin, P. (2017). Vision-based and marker-less surgical tool detection and tracking: a review of the literature. *Med Image Anal*, 35:633–654.

[Bouget et al., 2015] Bouget, D., Benenson, R., Omran, M., Riffaud, L., Schiele, B., and Jannin, P. (2015). Detecting Surgical Tools by Modelling Local Appearance and Global Shape. *IEEE Trans Med Imaging*, 34(12):2603–2617.

[Canziani et al., 2016] Canziani, A., Paszke, A., and Culurciello, E. (2016). An Analysis of Deep Neural Network Models for Practical Applications. *arXiv:1605.07678 [cs]*.

[Charrière et al., 2017] Charrière, K., Quellec, G., Lamard, M., Martiano, D., Cazuguel, G., Coatrieux, G., and Cochener, B. (2017). Real-time analysis of cataract surgery videos using statistical models. *Multimed Tools Appl*, 76(21):22473–22491.

[Chaudhry et al., 2009] Chaudhry, R., Ravichandran, A., Hager, G., and Vidal, R. (2009). Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1932–1939.

[Chen et al., 2013] Chen, C.-M., Chou, Y.-H., Tagawa, N., and Do, Y. (2013). Computer-Aided Detection and Diagnosis in Medical Imaging. *Comput Math Methods Med*, 2013.

[Choi et al., 2017] Choi, B., Jo, K., Choi, S., and Choi, J. (2017). Surgical-tools detection based on Convolutional Neural Network in laparoscopic robot-assisted surgery. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1756–1759.

[Cleary et al., 2005] Cleary, K., Kinsella, A., and Mun, S. (2005). Or 2020 workshop report: Operating room of the future. 1281:832–838.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA. IEEE Computer Society.

[Datta and Figueira, 2011] Datta, D. and Figueira, J. R. (2011). A real-integer-discrete-coded particle swarm optimization for design problems. *Applied Soft Computing*, 11(4):3625–3633.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

[Dergachyova et al., 2016] Dergachyova, O., Bouget, D., Huaulmé, A., Morandi, X., and Jannin, P. (2016). Automatic data-driven real-time segmentation and recognition of surgical workflow. *Int J Comput Assist Radiol Surg*, 11(6):1081–1089.

[Doebbeling et al., 2012] Doebbeling, B. N., Burton, M. M., Wiebke, E. A., Miller, S., Baxter, L., Miller, D., Alvarez, J., and Pekny, J. (2012). Optimizing Perioperative Decision Making: Improved Information for Clinical Workflow Planning. *AMIA Annu Symp Proc*, 2012:154–163.

[Doi, 2007] Doi, K. (2007). Computer-aided diagnosis in medical imaging: historical review, current status and future potential. *Comput Med Imaging Graph*, 31(4-5):198–211.

[Donahue et al., 2017] Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., and Darrell, T. (2017). Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):677–691.

[Doulamis et al., 2010] Doulamis, A., Doulamis, N., Kalisperakis, I., and Stentoumis, C. (2010). A real-time single-camera approach for automatic fall detection. *ISPRS Commission V, Close Range Image measurements Techniques*, 38:207–212.

[Droueche, 2012] Droueche, M. Z. (2012). *Fouille de séquences d'images médicales. Application en chirurgie mini-invasive augmentée.* phdthesis, Télécom Bretagne ; Université de Rennes 1.

[Farnebäck, 2003] Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer.

[Feichtenhofer et al., 2016] Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional Two-Stream Network Fusion for Video Action Recognition. *arXiv:1604.06573 [cs].* arXiv: 1604.06573.

[Forestier et al., 2015] Forestier, G., Riffaud, L., and Jannin, P. (2015). Automatic phase prediction from low-level surgical activities. *Int J Comput Assist Radiol Surg*, 10(6):833–841.

[Garcia-Peraza-Herrera et al., 2017] Garcia-Peraza-Herrera, L. C., Li, W., Fidon, L., Gruijthuijsen, C., Devreker, A., Attilakos, G., Deprest, J., Poorten, E. V., Stoyanov, D., Vercauteren, T., and Ourselin, S. (2017). ToolNet: Holistically-Nested Real-Time Segmentation of Robotic Surgical Tools. *arXiv:1706.08126 [cs].*

[García-Peraza-Herrera et al., 2016] García-Peraza-Herrera, L. C., Li, W., Gruijthuijsen, C., Devreker, A., Attilakos, G., Deprest, J., Poorten, E. V., Stoyanov, D., Vercauteren, T., and Ourselin, S. (2016). Real-Time Segmentation of Non-rigid Surgical Tools Based on Deep Learning and Tracking. In *Computer-Assisted and Robotic Endoscopy*, Lecture Notes in Computer Science, pages 84–95. Springer, Cham.

[Glaser et al., 2015] Glaser, B., Dänzer, S., and Neumuth, T. (2015). Intra-operative surgical instrument usage detection on a multi-sensor table. *Int J Comput Assist Radiol Surg*, 10(3):351–362.

[Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning.* MIT Press. `http://www.deeplearningbook.org`.

[Goyette et al., 2014] Goyette, N., Jodoin, P.-M., Porikli, F., Konrad, J., and Ishwar, P. (2014). A novel video dataset for change detection benchmarking. *IEEE Transactions on Image Processing*, 23(11):4663–4679.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151.

[Hashemi et al., 2018] Hashemi, F. S. G., Ismail, M. R., Yusop, M. R., Hashemi, M. S. G., Shahraki, M. H. N., Rastegari, H., Miah, G., and Aslani, F. (2018). Intelligent mining of large-scale bio-data: Bioinformatics applications. *Biotechnology & Biotechnological Equipment*, 32(1):10–29.

[He and Sun, 2015] He, K. and Sun, J. (2015). Convolutional neural networks at constrained time cost. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 5353–5360. IEEE.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[He et al., 2016a] He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[He et al., 2016b] He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Healthcare, 2008] Healthcare, F. . S. (2008). Prepare for disasters & tackle terabytes when evaluating medical image archiving.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

[Hospedales et al., 2012] Hospedales, T., Gong, S., and Xiang, T. (2012). Video Behaviour Mining Using a Dynamic Topic Model. *Int J Comput Vis*, 98(3):303–323.

[Hu et al., 2011] Hu, W., Xie, N., Li, L., Zeng, X., and Maybank, S. (2011). A Survey on Visual Content-Based Video Indexing and Retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):797–819.

[Hu et al., 2017] Hu, X., Yu, L., Chen, H., Qin, J., and Heng, P. (2017). Agnet: Attention-guided network for surgical tool presence detection. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support - Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, 2017, Proceedings*, pages 186–194.

[Huang et al., 2000] Huang, L., Chen, H., Wang, X., and Chen, G. (2000). A fast algorithm for mining association rules. *J. Comput. Sci. & Technol.*, 15(6):619–624.

[Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*.

[Jain and Vailaya, 1996] Jain, A. K. and Vailaya, A. (1996). Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244.

[Jin et al., 2016] Jin, Y., Dou, Q., Chen, H., Yu, L., and Heng, P.-A. (Oct 2016). Endorcn: Recurrent convolutional networks for recognition of surgical workflow in cholecystectomy procedure video. Technical report. The Chinese University of Hong Kong.

[Johnson et al., 2016] Johnson, J., Karpathy, A., and Fei-Fei, L. (2016). DenseCap: Fully Convolutional Localization Networks for Dense Captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4565–4574.

[Jolliffe, 2002] Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, New York, 2 edition.

[Kaggle, 2015] Kaggle (2015). Kaggle, diabetic retinopathy detection.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*.

[Klank et al., 2008] Klank, U., Padoy, N., Feussner, H., and Navab, N. (2008). Automatic feature generation in endoscopic images. *Int J CARS*, 3(3-4):331–339.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and E. Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 25.

[Lalys and Jannin, 2014] Lalys, F. and Jannin, P. (2014). Surgical process modelling: a review. *Int J Comput Assist Radiol Surg*, 9(3):495–511.

[Lalys et al., 2012] Lalys, F., Riffaud, L., Bouget, D., and Jannin, P. (2012). A Framework for the Recognition of High-Level Surgical Tasks From Video Images for Cataract Surgeries. *IEEE Transactions on Biomedical Engineering*, 59(4):966–976.

[Lalys et al., 2010] Lalys, F., Riffaud, L., Morandi, X., and Jannin, P. (2010). Surgical Phases Detection from Microscope Videos by Combining SVM and HMM. In *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging*, Lecture Notes in Computer Science, pages 54–62. Springer, Berlin, Heidelberg.

[Laptev, 2005] Laptev, I. (2005). On Space-Time Interest Points. *Int J Comput Vision*, 64(2-3):107–123.

[Lea et al., 2016] Lea, C., Reiter, A., Vidal, R., and Hager, G. D. (2016). Segmental Spatiotemporal CNNs for Fine-Grained Action Segmentation. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 36–52. Springer, Cham.

[Lecun et al., 1998] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Liao et al., 2012] Liao, S.-H., Chu, P.-H., and Hsiao, P.-Y. (2012). Data mining techniques and applications – A decade review from 2000 to 2011. *Expert Systems with Applications*, 39(12):11303–11311.

[Lin et al., 2006] Lin, H. C., Shafran, I., Yuh, D., and Hager, G. D. (2006). Towards automatic skill evaluation: detection and segmentation of robot-assisted surgical motions. *Comput. Aided Surg.*, 11(5):220–230.

[Lin et al., 2013] Lin, M., Chen, Q., and Yan, S. (2013). Network In Network. *arXiv:1312.4400 [cs]*.

[Lo et al., 2003] Lo, B. P. L., Darzi, A., and Yang, G.-Z. (2003). Episode Classification for the Analysis of Tissue/Instrument Interaction with Multiple Visual Cues. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003*, Lecture Notes in Computer Science, pages 230–237. Springer, Berlin, Heidelberg.

[Long et al., 2015] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

[Long et al., 2009] Long, L. R., Antani, S., Deserno, T. M., and Thoma, G. R. (2009). Content-Based Image Retrieval in Medicine. *Int J Healthc Inf Syst Inform*, 4(1):1–16.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

[Ma et al., 2017] Ma, C.-Y., Chen, M.-H., Kira, Z., and AlRegib, G. (2017). TS-LSTM and Temporal-Inception: Exploiting Spatiotemporal Dynamics for Activity Recognition. *arXiv:1703.10667 [cs]*.

[Malis and Vargas, 2007] Malis, E. and Vargas, M. (2007). Deeper understanding of the homography decomposition for vision-based control. report, INRIA.

[Manjunath and Ma, 1996] Manjunath, B. S. and Ma, W. Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842.

[Meißner and Neumuth, 2012] Meißner, C. and Neumuth, T. (2012). RFID-based surgical instrument detection using Hidden Markov models. 57.

[Mishra et al., 2017] Mishra, K., Sathish, R., and Sheet, D. (2017). Learning Latent Temporal Connectionism of Deep Residual Visual Abstractions for Identifying Surgical Tools in Laparoscopy Procedures. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2233–2240.

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

[Muja and Lowe, 2009] Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340.

[Otsu, 1979] Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

[Padhy et al., 2012] Padhy, N., Mishra, D. P., and Panigrahi, R. (2012). The Survey of Data Mining Applications And Feature Scope. *International Journal of Computer Science, Engineering and Information Technology*, 2(3):43–58. arXiv: 1211.5723.

[Padoy et al., 2012] Padoy, N., Blum, T., Ahmadi, S.-A., Feussner, H., Berger, M.-O., and Navab, N. (2012). Statistical modeling and recognition of surgical workflow. *Med Image Anal*, 16(3):632–641.

[Padoy et al., 2007] Padoy, N., Blum, T., Essa, I., Feussner, H., Berger, M.-O., and Navab, N. (2007). A Boosted Segmentation Method for Surgical Workflow Analysis. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, Lecture Notes in Computer Science, pages 102–109. Springer, Berlin, Heidelberg.

[Padoy et al., 2008] Padoy, N., Blum, T., Feussner, H., Berger, M.-O., and Navab, N. (2008). On-line Recognition of Surgical Activity for Monitoring in the Operating Room. In *Proceedings of the 20th National Conference on Innovative Applications of Artificial Intelligence - Volume 3*, IAAI'08, pages 1718–1724, Chicago, Illinois. AAAI Press.

[Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

[Parsons, 2005] Parsons, S. (2005). Independent Component Analysis: A Tutorial Introduction by James V. Stone, MIT Press. *Knowl. Eng. Rev.*, 20(2):198–199.

[Pernek and Ferscha, 2017] Pernek, I. and Ferscha, A. (2017). A survey of context recognition in surgery. *Medical & biological engineering & computing*.

[Pezzementi et al., 2009] Pezzementi, Z., Voros, S., and Hager, G. D. (2009). Articulated object tracking by rendering consistent appearance parts. In *2009 IEEE International Conference on Robotics and Automation*, pages 3940–3947.

[Piciarelli and Foresti, 2006] Piciarelli, C. and Foresti, G. L. (2006). On-line trajectory clustering for anomalous events detection. *Pattern Recognit. Lett*, pages 1835–1842.

[Quellec, 2008] Quellec, G. (2008). *Indexation et fusion multimodale pour la recherche d'information par le contenu. Application aux bases de données d'images médicales.* phdthesis, TELECOM Bretagne.

[Quellec et al., 2011] Quellec, G., Abramoff, M., Cazuguel, G., Lamard, M., Cochener, B., and Roux, C. (2011). Multiple-instance and multi-resolution image mining for diabetic retinopathy screening. *IRBM*, 32(6):342–350.

[Quellec et al., 2017a] Quellec, G., Cazuguel, G., Cochener, B., and Lamard, M. (2017a). Multiple-Instance Learning for Medical Image and Video Analysis. *IEEE Reviews in Biomedical Engineering*, PP(99):1–1.

[Quellec et al., 2017b] Quellec, G., Charrière, K., Boudi, Y., Cochener, B., and Lamard, M. (2017b). Deep image mining for diabetic retinopathy screening. *Medical Image Analysis*, 39:178–193.

[Quellec et al., 2014a] Quellec, G., Charrière, K., Lamard, M., Droueche, Z., Roux, C., Cochener, B., and Cazuguel, G. (2014a). Real-time recognition of surgical tasks in eye surgery videos. *Med Image Anal*, 18(3):579–590.

[Quellec et al., 2012a] Quellec, G., Lamard, M., Abràmoff, M. D., Decencière, E., Lay, B., Erginay, A., Cochener, B., and Cazuguel, G. (2012a). A multiple-instance learning framework for diabetic retinopathy screening. *Medical Image Analysis*, 16(6):1228–1240.

[Quellec et al., 2010a] Quellec, G., Lamard, M., Cazuguel, G., Cochener, B., and Roux, C. (2010a). Adaptive Nonseparable Wavelet Transform via Lifting and its Application to Content-Based Image Retrieval. *IEEE Transactions on Image Processing*, 19(1):25–35.

[Quellec et al., 2010b] Quellec, G., Lamard, M., Cazuguel, G., Cochener, B., and Roux, C. (2010b). Wavelet optimization for content-based image retrieval in medical databases. *Medical Image Analysis*, 14(2):227–241.

[Quellec et al., 2012b] Quellec, G., Lamard, M., Cazuguel, G., Cochener, B., and Roux, C. (2012b). Fast Wavelet-Based Image Characterization for Highly Adaptive Image Retrieval. *IEEE Transactions on Image Processing*, 21(4):1613–1623.

[Quellec et al., 2014b] Quellec, G., Lamard, M., Cochener, B., and Cazuguel, G. (2014b). Real-time segmentation and recognition of surgical tasks in cataract surgery videos. *IEEE Trans Med Imaging*, 33(12):2352–2360.

[Quellec et al., 2015] Quellec, G., Lamard, M., Cochener, B., and Cazuguel, G. (2015). Real-time task recognition in cataract surgery videos using adaptive spatiotemporal polynomials. *IEEE Trans Med Imaging*, 34(4):877–887.

[Quellec et al., 2016a] Quellec, G., Lamard, M., Cozic, M., Coatrieux, G., and Cazuguel, G. (2016a). Multiple-Instance Learning for Anomaly Detection in Digital Mammography. *IEEE Transactions on Medical Imaging*, 35(7):1604–1614.

[Quellec et al., 2016b] Quellec, G., Lamard, M., Erginay, A., Chabouis, A., Massin, P., Cochener, B., and Cazuguel, G. (2016b). Automatic detection of referral patients due to retinal pathologies through data mining. *Medical Image Analysis*, 29(Supplement C):47–64.

[Quellec et al., 2008] Quellec, G., Lamard, M., Josselin, P. M., Cazuguel, G., Cochener, B., and Roux, C. (2008). Optimal wavelet transform for the detection of microaneurysms in retina photographs. *IEEE Trans Med Imaging*, 27(9):1230–1241.

[Raju et al., 2016] Raju, A., Wang, S., and Huang, J. (Oct 2016). "m2cai surgical tool detection challenge report," university of texas at arlington, tech. rep. Technical report.

[Reiley and Hager, 2009] Reiley, C. E. and Hager, G. D. (2009). Task versus subtask surgical skill evaluation of robotic minimally invasive surgery. *Med Image Comput Comput Assist Interv*, 12(Pt 1):435–442.

[Reiley et al., 2011] Reiley, C. E., Lin, H. C., Yuh, D. D., and Hager, G. D. (2011). Review of methods for objective surgical skill evaluation. *Surg Endosc*, 25(2):356–366.

[Reiter et al., 2012] Reiter, A., Allen, P. K., and Zhao, T. (2012). Marker-less articulated surgical tool detection. In *Computer assisted radiology and surgery*.

[Reiter et al., 2014] Reiter, A., Allen, P. K., and Zhao, T. (2014). Appearance learning for 3d tracking of robotic surgical tools. *The International Journal of Robotics Research*, 33(2):342–356.

[Reiter et al., 2011] Reiter, A., Goldman, R. E., Bajo, A., Iliopoulos, K., Simaan, N., and Allen, P. K. (2011). A learning algorithm for visual pose estimation of continuum robots. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2390–2396.

[Rieke et al., 2016] Rieke, N., Tan, D. J., Amat di San Filippo, C., Tombari, F., Alsheakhali, M., Belagiannis, V., Eslami, A., and Navab, N. (2016). Real-time localization of articulated surgical instruments in retinal microsurgery. *Med Image Anal*, 34:82–100.

[Roy et al., 2016] Roy, N., Misra, A., and Cook, D. (2016). Ambient and smartphone sensor assisted ADL recognition in multi-inhabitant smart environments. *J Ambient Intell Human Comput*, 7(1):1–19.

[Rumelhart and McClelland, 1987] Rumelhart, D. E. and McClelland, J. L. (1987). Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing:Explorations in the Microstructure of Cognition: Foundations*, pages 318–362. MIT Press.

[Sahu et al., 2016] Sahu, M., Mukhopadhyay, A., Szengel, A., and Zachow, S. (2016). Tool and Phase recognition using contextual CNN features. *arXiv:1610.08854 [cs]*.

[Sarikaya et al., 2017] Sarikaya, D., Corso, J. J., and Guru, K. A. (2017). Detection and Localization of Robotic Tools in Robot-Assisted Surgery Videos Using Deep Neural Networks for Region Proposal and Detection. *IEEE Transactions on Medical Imaging*, 36(7):1542–1549.

[Saxe et al., 2013] Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:1312.6120 [cond-mat, q-bio, stat]*.

[Schuster and Paliwal, 1997] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

[Shen et al., 2017] Shen, D., Wu, G., and Suk, H.-I. (2017). Deep Learning in Medical Image Analysis. *Annu Rev Biomed Eng*, 19:221–248.

[Shin et al., 2015] Shin, H. J., Kim, H. H., and Cha, J. H. (2015). Current status of automated breast ultrasonography. *Ultrasonography*, 34(3):165–172.

[Simonyan and Zisserman, 2014a] Simonyan, K. and Zisserman, A. (2014a). Two-stream Convolutional Networks for Action Recognition in Videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 568–576, Cambridge, MA, USA. MIT Press.

[Simonyan and Zisserman, 2014b] Simonyan, K. and Zisserman, A. (2014b). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*.

[Specht, 1988] Specht, D. F. (1988). Probabilistic neural networks for classification, mapping, or associative memory. In *IEEE 1988 International Conference on Neural Networks*, pages 525–532 vol.1.

[Srivastava et al., 2015] Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Highway networks. *arXiv preprint arXiv:1505.00387*.

[Sun et al., 2015] Sun, L., Jia, K., Yeung, D.-Y., and Shi, B. E. (2015). Human Action Recognition Using Factorized Spatio-Temporal Convolutional Networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4597–4605, Washington, DC, USA. IEEE Computer Society.

[Suzuki et al., 2015] Suzuki, T., Egi, H., Hattori, M., Tokunaga, M., Sawada, H., and Ohdan, H. (2015). An evaluation of the endoscopic surgical skills assessment using a video analysis software program. *Surg Endosc*, 29(7):1804–1808.

[Szegedy et al., 2016a] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2016a). Inception-v4, inception-resnet and the impact of residual connections on learning. In *ICLR 2016 Workshop*.

[Szegedy et al., 2015a] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015a). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

[Szegedy et al., 2015b] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015b). Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Szegedy et al., 2016b] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016b). Rethinking the Inception Architecture for Computer Vision. pages 2818–2826.

[Szegedy et al., 2016c] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016c). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

[Teynor, 2006] Teynor, A. (2006). Patch based approaches for visual object class recognition-a survey.

[Tran et al., 2015] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. *IEEE Int. Conf. Comput. Vis*, pages 4489–4497.

[Trikha et al., 2013] Trikha, S., Turnbull, A. M. J., Morris, R. J., Anderson, D. F., and Hossain, P. (2013). The journey to femtosecond laser-assisted cataract surgery: new beginnings or a false dawn? *Eye (Lond)*, 27(4):461–473.

[Twinanda et al., 2015] Twinanda, A. P., Marescaux, J., de Mathelin, M., and Padoy, N. (2015). Classification approach for automatic laparoscopic video database organization. *Int J Comput Assist Radiol Surg*, 10(9):1449–1460.

[Twinanda et al., 2016] Twinanda, A. P., Mutter, D., Marescaux, J., de Mathelin, M., and Padoy, N. (2016). Single- and Multi-Task Architectures for Surgical Workflow Challenge at M2cai 2016. *arXiv:1610.08844 [cs]*.

[Twinanda et al., 2017] Twinanda, A. P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M., and Padoy, N. (2017). EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos. *IEEE Trans Med Imaging*, 36(1):86–97.

[V et al., 2016] V, G., L, P., M, C., and et al (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410.

[Voros et al., 2007] Voros, S., Long, J.-A., and Cinquin, P. (2007). Automatic Detection of Instruments in Laparoscopic Images: A First Step Towards High-level Command of Robotic Endoscopic Holders. *The International Journal of Robotics Research*, 26(11-12):1173–1190.

[Wang et al., 2017] Wang, S., Raju, A., and Huang, J. (2017). Deep learning based multi-label classification for surgical tool presence detection in laparoscopic videos. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 620–623.

[Werbos, 1990] Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

[Wu et al., 2007] Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., and Rehg, J. M. (2007). A Scalable Approach to Activity Recognition based on Object Use. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8.

[Xia and Aggarwal, 2013] Xia, L. and Aggarwal, J. K. (2013). Spatio-temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 2834–2841, Washington, DC, USA. IEEE Computer Society.

[Xie and Tu, 2015] Xie, S. and Tu, Z. (2015). Holistically-Nested Edge Detection. *arXiv:1504.06375 [cs]*.

[Xu et al., 2012] Xu, R., Agarwal, P., Kumar, S., Krovi, V. N., and Corso, J. J. (2012). Combining Skeletal Pose with Local Motion for Human Activity Recognition. In *Articulated Motion and Deformable Objects*, Lecture Notes in Computer Science, pages 114–123. Springer, Berlin, Heidelberg.

[Xu et al., 2016] Xu, Z., Hu, C., and Mei, L. (2016). Video structured description technology based intelligence analysis of surveillance videos for public security applications. *Multimed Tools Appl*, 75(19):12155–12172.

[Yue-Hei Ng et al., 2015] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., and Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. pages 4694–4702.

[Zappella et al., 2013] Zappella, L., Béjar, B., Hager, G., and Vidal, R. (2013). Surgical gesture classification from video and kinematic data. *Medical Image Analysis*, 17(7):732–745.

[Zhu et al., 2016] Zhu, W., Hu, J., Sun, G., Cao, X., and Qiao, Y. (2016). A Key Volume Mining Deep Framework for Action Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1991–1999.

[Zia et al., 2016] Zia, A., Castro, D., and Essa, I. (Oct 2016). "fine-tuning deep architectures for surgical tool detection," georgia institute of technology, tech. rep. Technical report.

[Zoph and Le, 2016] Zoph, B. and Le, Q. V. (2016). Neural Architecture Search with Reinforcement Learning. *arXiv:1611.01578 [cs]*.

[Zoph et al., 2017] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2017). Learning Transferable Architectures for Scalable Image Recognition. *arXiv:1707.07012 [cs, stat]*.

9
Appendices

:

A Homography Experimentations

The *homography* transformation is a projective transformation: two images of the-same planar object taken from two different points of view are linked by this transformation. The idea is to be able to recognize each tool on the surgical tray, relying on a dataset of images for the surgical tools (called tools reference images I_r) acquired from an angle of view different than the one used in the OR. Given the projection of a point of the tool in I_r, it is possible to determine where this point is projected on the real scene images, referred as I_s.

We denote by $X_i(x_i, y_i, 1)$ the homogeneous coordinates of a point in I_r and by $X'_i(x'_i, y'_i, 1)$ the homogeneous coordinates of a point in I_s. The projective transformation can be expressed as follows:

$$\alpha X'_i = HX_i \quad / \quad H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{9.1}$$

H can be measured by matching several coplanar points, up to a scale factor α. At least 4 points are needed. Fundamentally, the *homography* transformation can be expressed in terms of rotation and translation between camera coordinates, as shown in Figure 9.1.

$$H = R + \frac{1}{d} \times T \cdot N \tag{9.2}$$

where R is the rotation matrix from the camera coordinate F to other camera coordinate F^*, T is the translation matrix from the camera coordinate F to other camera coordinate F^*. d is the distance from center of F to the planar object.

N is the normal vector of the planar object. For a comprehensive review of the homography transformation, we refer the reader to [Agarwal et al., 2005].

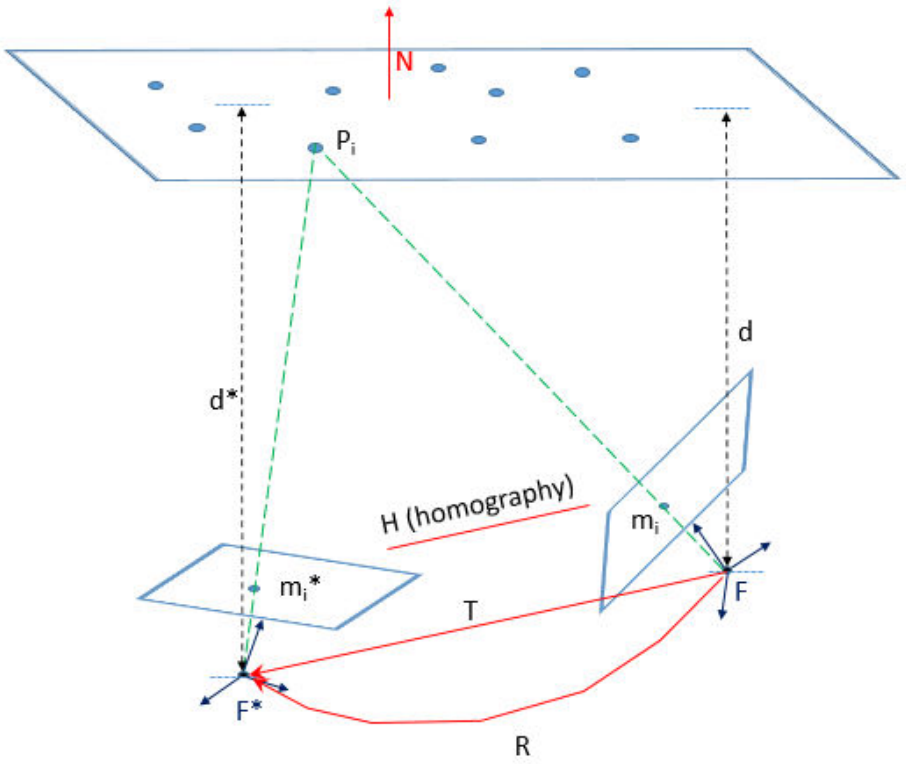


Figure 9.1: Homography transformation decomposition. Courtsy to [Malis and Vargas, 2007]

To obtain the feature representation of I_s and I_r, we extract visual features from the images. The visual features belong to one feature groups: interest points.
Regions of Tools. In order to focus on the object's regions of interest, we apply a *Sobel* edge detector which performs a 2-D spatial gradient measurement on images. Thereafter, we identify the connected components in the images where each component potentially corresponding to a tool. In I_r, there is only one connected component corresponding to the targeted tool. In I_s, each component can be a surgical tool or any other object laid down on the tray.
Points of Interest. We extract the speeded-up robust features (SURF) [Bay et al., 2006] because of its scale- and rotation-invariant characteristics. Each of the detected SURF key points are described with 128-dimensional feature vector.
Matching Points. To match the points extracted from I_r and the points of each component in I_s, we use a fast approximate of nearest neighbors [Muja and Lowe, 2009].

The experiments using this transformation were merely conducted on a sample of tray images. The *homography* transformation H is computed on the tool con-

nected component in I_r in regard to each connected component in the I_s. In fact, it uses the list of matched points between these two components to compute H (see equation (9.1)). As shown in Fig 9.2, applying the homography transformation to the reference image of the viscoelastic canula did not work well: it is rarely possible to find the reference tools in the real scene images.

Figure 9.2: A sample of I_r is on the left and a sample of I_s is on the right. The yellow box is the bounding box for the targeted tool connected component. The white points are the key-points in the reference tool image and inside the bounding box. The red circle surrounds the actual result of applying H on the corner points of I_r.

The problem boils down predominantly to one of the following reasons: (1) the key-points may not be very well suited for this type of problems due to the inability of covering all possible tool's viewpoints. (2) the criteria of matching the key-points may not be accurate. (3) the *homography* is unlikely to tolerate non-planar objects. In other words, the constraints, non-planar objects, the complexity of finding representative points of interest and expected noise on the data, adversely affects the results obtained by this method. The effectiveness of this method was very limited under the required constraints. Therefore, the *homography* transformation is inadequate to handle the surgical tool detection on the tray. Furthermore, the complexity of the problem and the inherent properties of the tools (low distinctive patterns, thin and small tools etc...), which may imply adapting the *homography* to each tool or to each category of tools separately, impede a practical solution based on the *homography* transformation.

B Results Of Surgical Tool Presence Detection

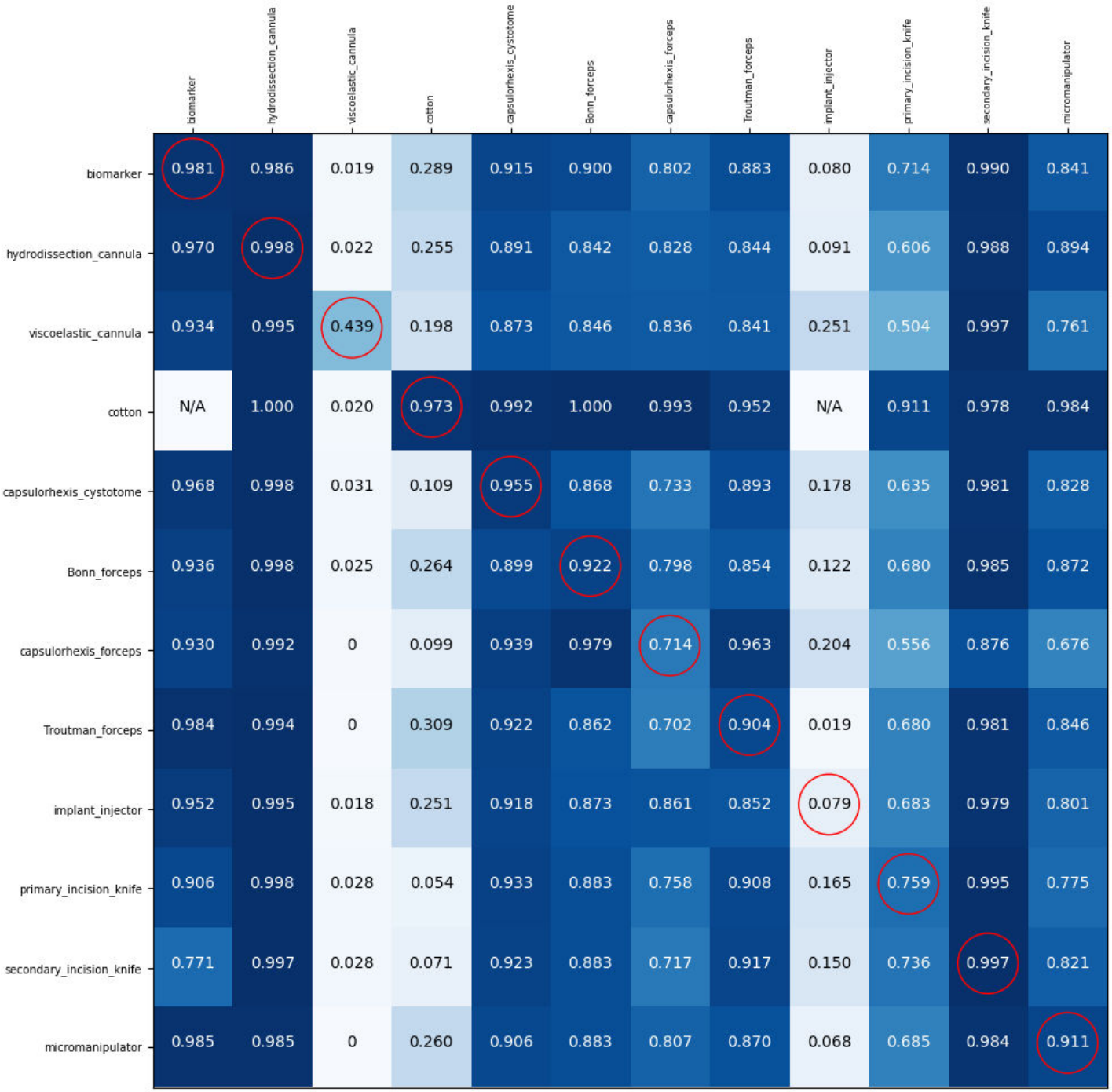


Figure 9.3: Confusion matrix for Inception-ResNet-V2 (P-CNN) tool absence detection of the evaluation using of the RW testing subset. For easier understanding, the diagonal cells are circled in red. N/A is not applicable: no images were found where the class in row is absent and the class in column is present.

Tool	ResNet-152 (P-CNN)	Inception-V4 (P-CNN)	Inception-ResNet-V2 (P-CNN)
biomarker	0.676	0.47	**0.763**
Charleux cannula	0.388	**0.554**	0.415
hydrodissection cannula	0.787	0.837	**0.899**
Rycroft cannula	0.643	0.81	**0.832**
viscoelastic cannula	0.959	0.961	**0.973**
cotton	**0.59**	0.259	0.381
capsulorhexis cystotome	**0.897**	0.824	0.761
Bonn forceps	0.726	0.79	**0.848**
capsulorhexis forceps	0.738	0.846	**0.878**
Troutman forceps	0.615	0.762	**0.783**
needle holder	0.421	**0.605**	0.576
irrigation/aspiration handpiece	**0.941**	0.924	0.871
phacoemulsifier handpiece	0.285	0.203	**0.375**
vitrectomy handpiece	0.671	0.45	**0.744**
implant injector	**0.705**	0.657	0.689
primary incision knife	0.84	**0.93**	0.908
secondary incision knife	0.878	0.886	**0.896**
micromanipulator	0.863	**0.931**	0.856
suture needle	0.399	**0.497**	0.488
Mendez ring	**0.88**	0.484	0.685
Vannas scissors	0.432	0.491	**0.512**
Average (mA_z)	0.638	0.675	**0.721**
Standard deviation	0.201	0.228	0.188

Table 9.1: P-CNN results in terms of areas under the ROC curve (A_z) for surgical tray videos. For each tool, the highest score is marked in bold.

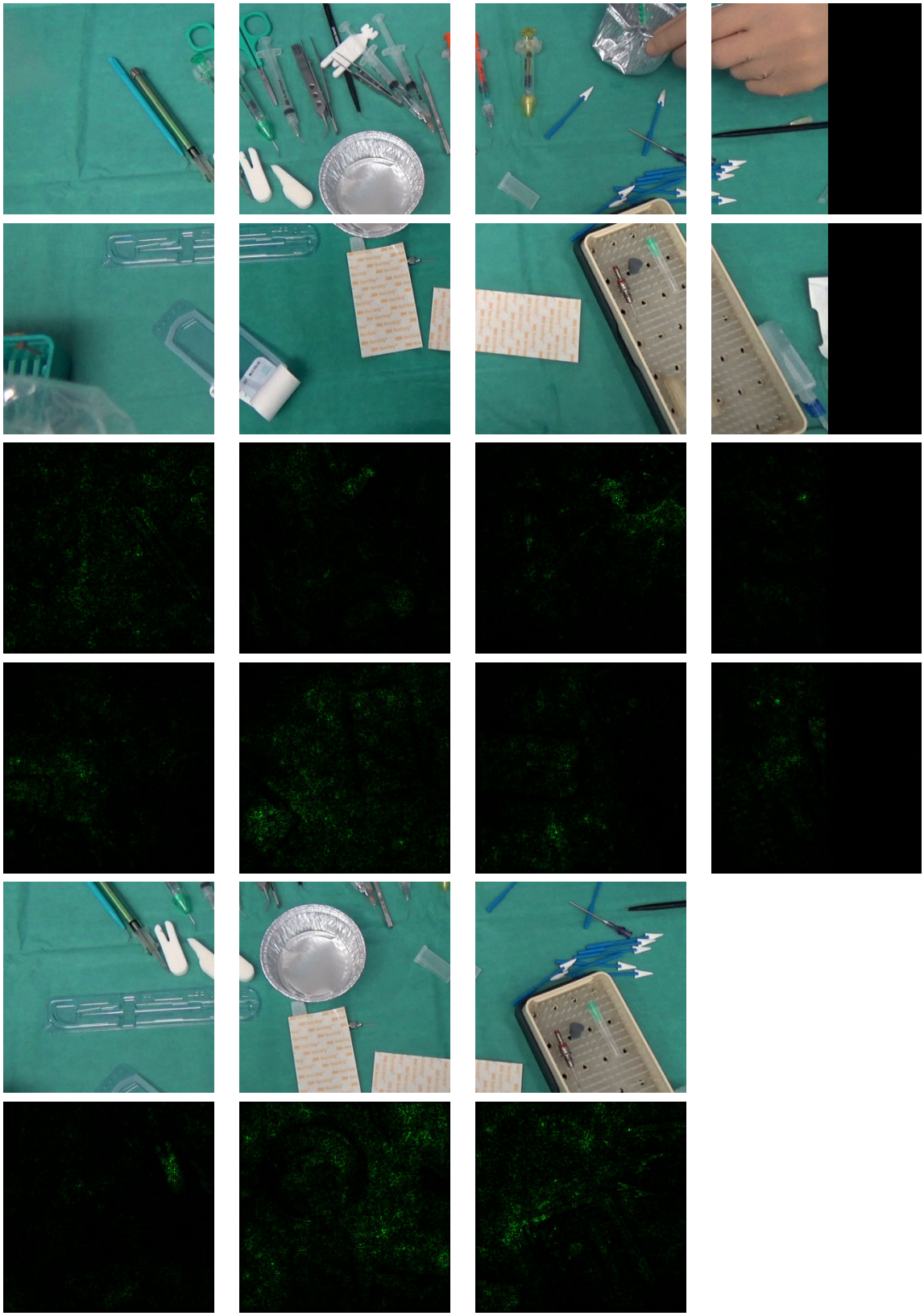


Figure 9.4: Hue-constrained sensitivity analysis for best performing network using P-CNN: Inception-ResNet-V2. These examples were taken from the testing set of the surgical tray videos.

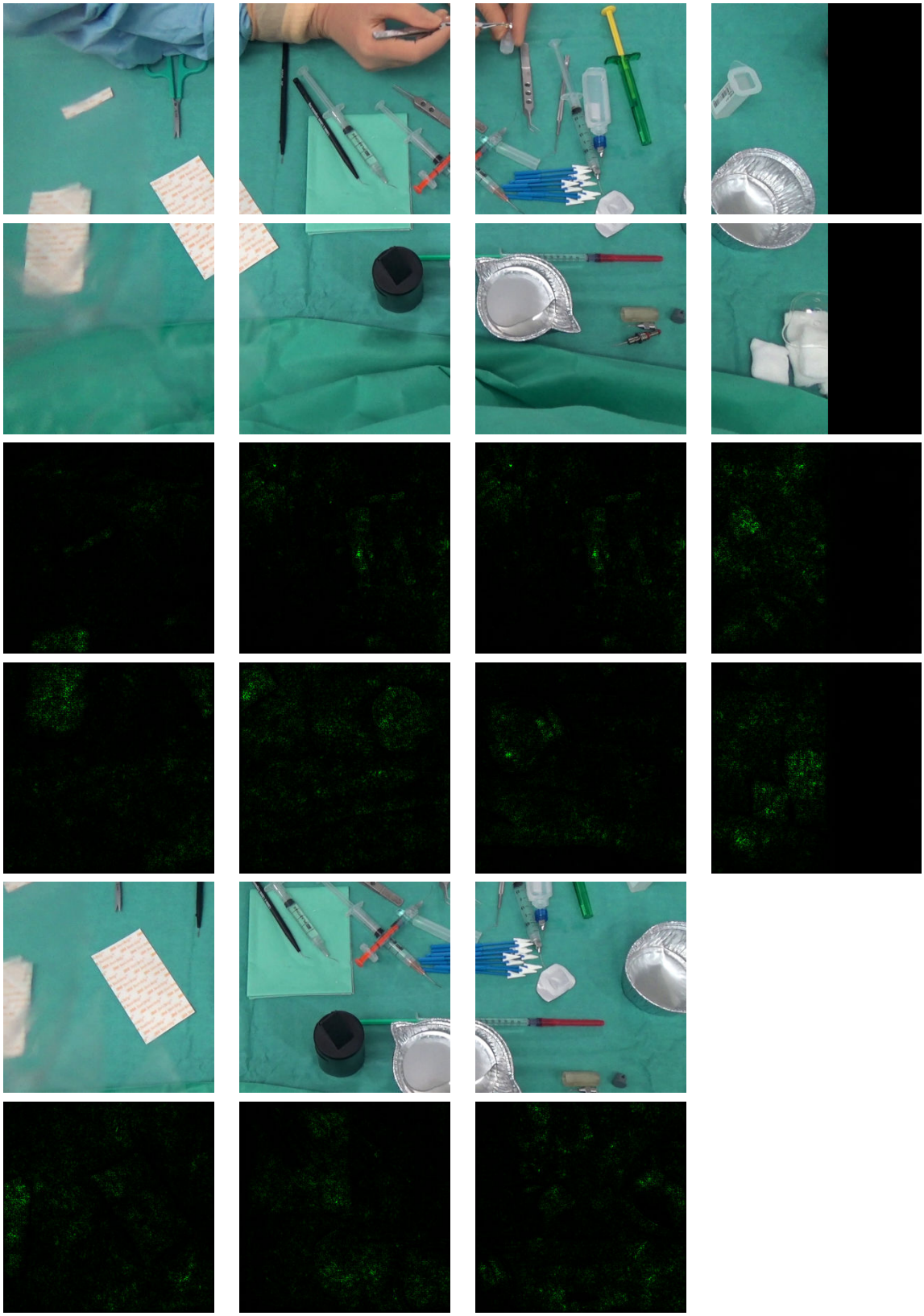


Figure 9.5: Hue-constrained sensitivity analysis for best performing network using P-CNN: Inception-ResNet-V2. These examples were taken from the testing set of the surgical tray videos.

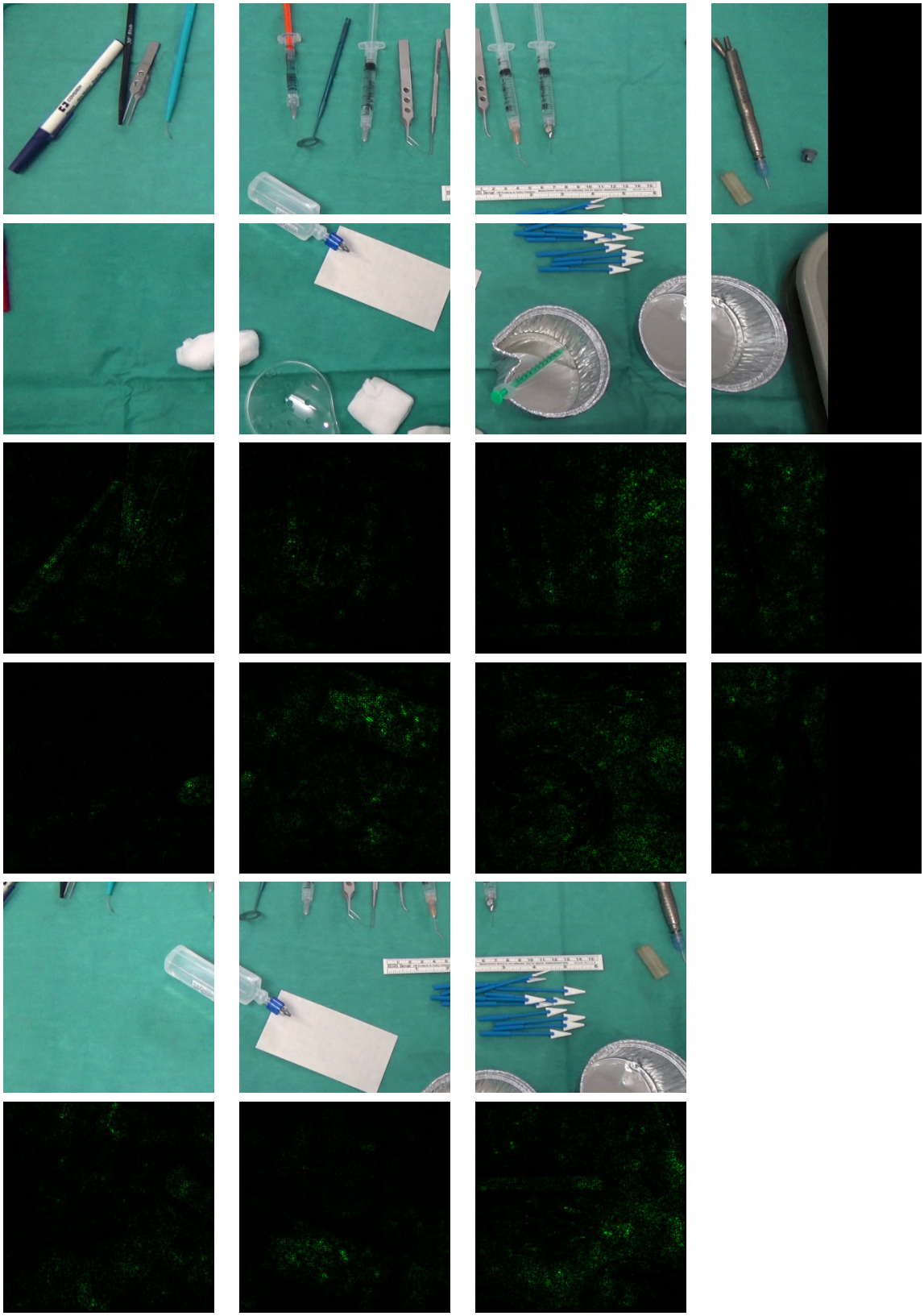


Figure 9.6: Hue-constrained sensitivity analysis for best performing network using P-CNN: Inception-ResNet-V2. These examples were taken from the testing set of the surgical tray videos.

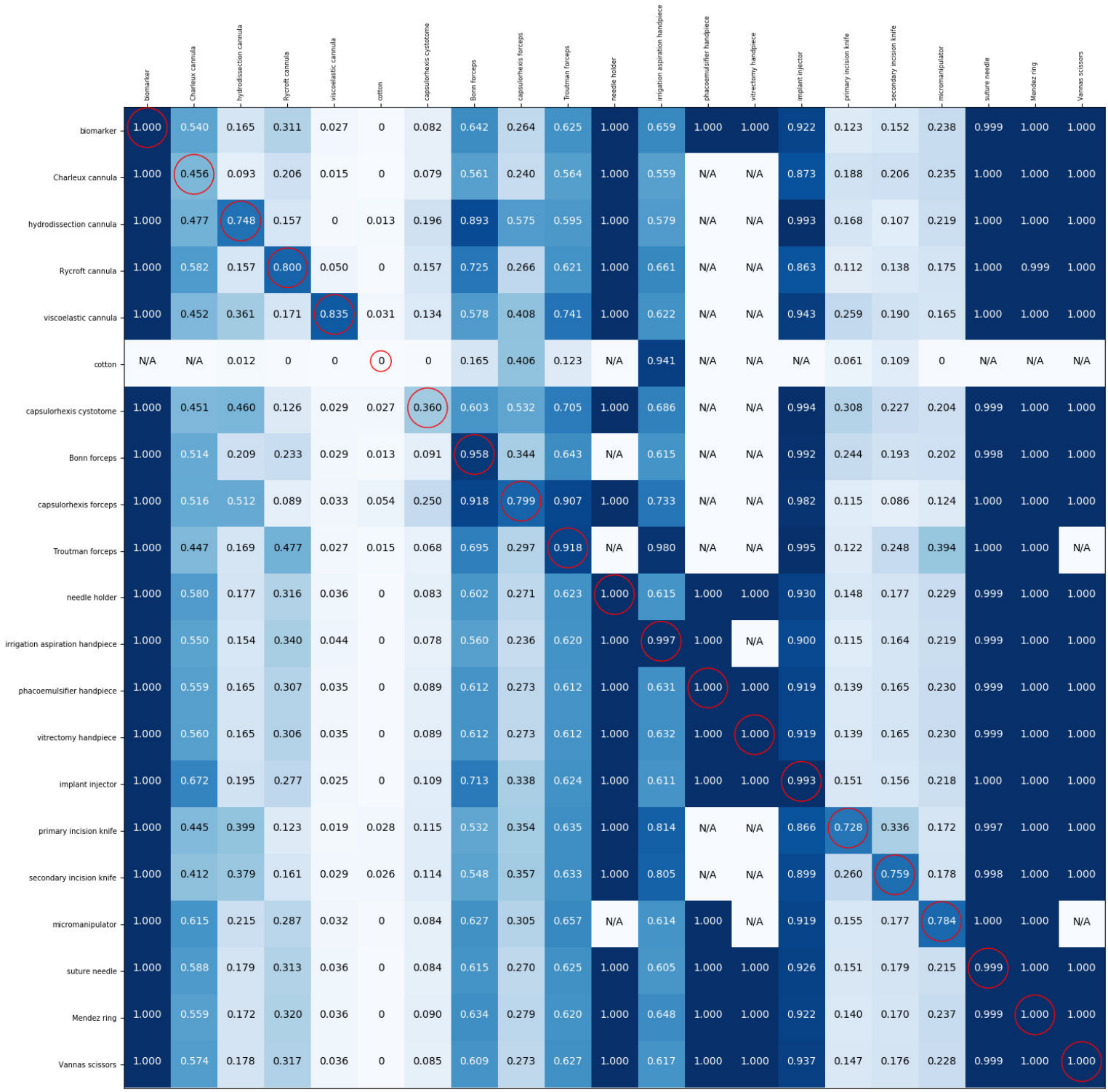


Figure 9.7: Confusion matrix for Inception-ResNet-V2 (P-CNN) tool absence (no presence) detection. For easier understanding, the diagonal cells are circled in red. N/A is not applicable: no images were found where the class in row is absent and the class in column is present.

C Publications Related to This Thesis

Here, we show two papers submitted to Medical Image Analysis. One presents a boosting methodology on top of the "CNN+RNN" applied on the tool-tissue interaction videos. The second is the outcome of CATARACTS 2017, a paper summarizing the top ranking solutions, which has been submitted for review. A third paper, published in EMBC 2017, is presented, in which we exploit the optical flow inside the CNN.

Monitoring Tool Usage in Surgery Videos using Boosted Convolutional and Recurrent Neural Networks

Hassan Al Hajj[a], Mathieu Lamard[b,a], Pierre-Henri Conze[c,a], Béatrice Cochener[b,a,d], Gwenolé Quellec[a,*]

[a]*Inserm, UMR 1101, Brest, F-29200 France*
[b]*Univ Bretagne Occidentale, Brest, F-29200 France*
[c]*Institut Mines-Télécom Atlantique, Brest, F-29200 France*
[d]*Service d'Ophtalmologie, CHRU Brest, Brest, F-29200 France*

Abstract

This paper investigates the automatic monitoring of tool usage during a surgery, with potential applications in report generation, surgical training and real-time decision support. Two surgeries are considered: cataract surgery, the most common surgical procedure, and cholecystectomy, one of the most common digestive surgeries. Tool usage is monitored in videos recorded either through a microscope (cataract surgery) or an endoscope (cholecystectomy). Following state-of-the-art video analysis solutions, each frame of the video is analyzed by convolutional neural networks (CNNs) whose outputs are fed to recurrent neural networks (RNNs) in order to take temporal relationships between events into account. Novelty lies in the way those CNNs and RNNs are trained. Computational complexity prevents the end-to-end training of "CNN+RNN" systems. Therefore, CNNs are usually trained first, independently from the RNNs. This approach is clearly suboptimal for surgical tool analysis: many tools are very similar to one another, but they can generally be differentiated based on past events. CNNs should be trained to extract the most useful visual features in combination with the temporal context. A novel boosting strategy is proposed to achieve this

*LaTIM - IBRBS - CHRU Morvan - 12, Av. Foch
29609 Brest CEDEX - FRANCE
Tel.: +33 2 98 01 81 29 / Fax: +33 2 98 01 81 24
Email address: gwenole.quellec@inserm.fr (Gwenolé Quellec)

goal: the CNN and RNN parts of the system are simultaneously enriched by progressively adding weak classifiers (either CNNs or RNNs) trained to improve the overall classification accuracy. Experiments were performed in a dataset of 50 cataract surgery videos, where the usage of 21 surgical tools was manually annotated, and a dataset of 80 cholecystectomy videos, where the usage of 7 tools was manually annotated. Very good classification performance are achieved in both datasets: tool usage could be labeled with an average area under the ROC curve of $A_z = 0.9961$ and $A_z = 0.9939$, respectively, in offline mode (using past, present and future information), and $A_z = 0.9957$ and $A_z = 0.9936$, respectively, in online mode (using past and present information only).

Keywords: cataract and cholecystectomy surgeries, tool usage monitoring, video analysis, Convolutional and Recurrent Neural Networks, boosting

1. Introduction

With the emergence of imaging devices in the operating room, the automated analysis of videos recorded during the surgery is becoming a hot research topic. In particular, videos can be used to monitor the surgery, for instance by recognizing which surgical tools are being used at every moment. An immediate application of surgery monitoring is report generation. If automatic reports are available for many surgeries, then the automatic analysis of these reports can help optimize the surgical workflow or evaluate surgical skills. Additionally, if we are able to generate such a report in real-time, during a surgery, then we could compare it with previous reports to generate warnings, if we recognize patterns often leading to complications, or recommendations, to help younger surgeons emulate more experienced colleagues based on their surgical reports [Quellec et al., 2014][Quellec et al., 2015]. With adequate image analysis techniques, tool usage could be monitored reliably in tool-interaction videos, such as endoscopic videos (in laparoscopic or retinal surgeries) or microscopic videos (in anterior eye segment surgeries). In the simplest scenario, we can consider that a tool is being used if it is visible in these videos. In a more advanced scenario, we can consider that it is in use if it is in contact with the tissue (as opposed to approaching the tissue, waiting to be used, etc.). Therefore, several tool detection techniques for tool-interaction videos have been proposed in recent years [Bouget et al., 2017]. To compare these techniques, two tool detection challenges were or-

ganized recently. A first challenge, organized at the M2CAI 2016 workshop,[1] relied on endoscopic videos of cholecystectomy operations performed laparoscopically. We organized a second challenge for cataract surgery, the most common surgical procedure worldwide [Trikha et al., 2013].[2] It relied on videos recorded through a surgical microscope. Following the trend in medical image and video analysis [Shen et al., 2017], the best solutions of both challenges all relied on convolutional neural networks (CNNs) [Raju et al., 2016; Sahu et al., 2016; Twinanda et al., 2017; Zia et al., 2016; Roychowdhury et al., 2017; Hu and Heng, 2017; Maršalkaitė et al., 2017].

Compared to other computer vision tasks, surgical tool usage annotation has several specificities. First, as opposed to many computer vision tasks, including the popular ImageNet visual recognition challenges,[3] the problem at hand is not multiclass classification (one correct label per image among multiple classes), but rather multilabel classification (multiple correct labels per image): the number of tools being used in each image varies (from zero to three in cataract surgery for instance). Therefore, multilabel CNNs should be used. Second, taking the temporal sequencing into account is important: knowing which tools have already been used since the beginning of the surgery greatly helps recognize which tools are currently being used. Therefore, multilabel recurrent neural networks (RNNs) [Hochreiter and Schmidhuber, 1997] may also be used advantageously. In fact, recent machine learning competitions clearly show that ensembles of CNNs outperform single CNNs [Russakovsky et al., 2015]: multiple CNNs with different architectures are generally trained independently and their outputs are combined afterward using standard machine learning algorithms (decision trees, random forests, multilayer perceptrons, etc.). However, this simple strategy is suboptimal since difficult samples may be misclassified by all CNNs. And there are many difficult samples to classify in surgery videos: in particular, many tools resemble one other (e.g. two types of cannulae in cataract surgery). Building the ensemble of CNNs using a boosting meta-algorithm [Freund and Schapire, 1997] can theoretically design CNNs focusing specifically on challenging samples. Boosting an ensemble of RNNs would also make sense as there are difficult samples along the time dimension as well:

[1] `http://camma.u-strasbg.fr/m2cai2016/index.php/tool-presence-detection-challenge-results/`

[2] `https://cataracts.grand-challenge.org/`

[3] `http://www.image-net.org/challenges/LSVRC/2017/index.php`

in particular, some tools or tool usage sequences are very rare and temporal sequencing algorithms tend to misclassify those rare cases. Therefore, we propose to jointly boost an ensemble of CNNs and an ensemble of RNNs for automatic tool usage annotation in surgery videos. In the same way as CNN boosting (or RNN boosting) allows various CNNs (or RNNs) to be complementary, this general boosting solution allows CNNs to be complementary with RNNs. In that sense, it approximates the end-to-end training of a "CNN+RNN" network, which is theoretically ideal but not computationally tractable.

The remainder of this paper is organized as follows. Section 2 reviews the state of the art of video analysis, and surgery video analysis in particular. Sections 3 and 4 describe the proposed solution. Section 5 presents the video datasets and section 6 reports the experiments performed on that dataset. We end with a discussion and conclusions in section 7.

2. State of the Art

2.1. Deep Learning for Video Analysis

The automatic analysis of dynamic scenes through deep learning has become a very hot research topic [Simonyan and Zisserman, 2014; Wang et al., 2017; Donahue et al., 2017]. Different strategies have been proposed for this task. A first strategy is to regard videos or video portions as 3-D images and therefore analyze them with 3-D CNNs [Ji et al., 2013], which is computationally expansive. A second strategy is to analyze 2-D images as well as the optical flow between consecutive images [Simonyan and Zisserman, 2014], with the disadvantage of only modeling short-term relationships between images. A third strategy is to combine a CNN, analyzing 2-D images, with a RNN analyzing the temporal sequencing [Donahue et al., 2017]. The main advantage of this "CNN+RNN" approach, which is now the leading video analysis solution, is that long-term relationships between events can be taken into account efficiently. One application of "CNN+RNN" models, which is particularly relevant for our study, is video labeling: the goal is to assign one class label to each frame inside a video [Singh et al., 2016; Khorrami et al., 2016]. Medical applications of this research, ranging from gait analysis [Feng et al., 2016] to surgery monitoring [Bodenstedt et al., 2017; Twinanda et al., 2016], are starting to emerge.

2.2. Temporal Analysis of Surgery Videos

In the context of surgical workflow analysis, solutions have been proposed to recognize surgical phases in surgery videos [Lalys and Jannin, 2014; Charrière et al., 2017]. In Primus et al. [2018], phases are recognized using one CNN processing the visual content of one frame plus the relative timestamp of that frame. However, most solutions rely on statistical models, such as Hidden Markov Models (HMMs) [Cadène et al., 2016], Hidden semi-Markov Models [Dergachyova et al., 2016; Tran et al., 2017], Hierarchical HMMs [Twinanda et al., 2017], Linear Dynamical Systems [Zappella et al., 2013; Tran et al., 2017] or Conditional Random Fields [Tao et al., 2013; Quellec et al., 2014; Lea et al., 2016a]. Recently, solutions based on RNNs have also been proposed [Jin et al., 2016; Bodenstedt et al., 2017; Twinanda et al., 2016]. Following the state-of-the-art video analysis strategy, these RNNs process instant visual features extracted by a CNN from images. In particular, Jin et al. [2016] applied a "CNN+RNN" network to a small sliding window of three images. Bodenstedt et al. [2017] applied a "CNN+RNN" network to larger sliding windows and copy the internal state of the network between consecutive window locations. As for Twinanda et al. [2016], they applied a "CNN+RNN" network to full videos. Interestingly, the CNN proposed by Twinanda et al. [2016], namely EndoNet, detects tools as an intermediate step. A challenge on surgical workflow analysis was also organized at M2CAI 2016:[4] two of the top three solutions relied on RNNs [Jin et al., 2016; Twinanda et al., 2016]. It should be noted that successful works on the analysis of kinematics surgery data have also been reported, using a RNN [Dipietro et al., 2016] or a CNN along the temporal dimension [Lea et al., 2016b]. In all these works, statistical models or RNNs were used to label surgical activities and phases. Given the strong correlation between surgical activities and tool usage, they can be expected to improve tool recognition as well.

2.3. Deep Learning for Surgical Tool Detection

As evidenced by the M2CAI 2016 and CATARACTS 2017 challenges, the state-of-the-art algorithms for tool detection in surgery videos are CNNs. The best solutions of these challenges rely on a transfer learning strategy:

[4] `http://camma.u-strasbg.fr/m2cai2016/index.php/workflow-challenge-results/`

well-known CNNs trained to classify still images in the ImageNet dataset were fine-tuned on images extracted from surgery videos. For M2CAI 2016, Sahu et al. [2016] and Twinanda et al. [2017] fine-tuned AlexNet [Krizhevsky et al., 2012], Raju et al. [2016] fine-tuned GoogleNet [Szegedy et al., 2015a] and VGG-16 [Simonyan and Zisserman, 2015], and Zia et al. [2016] fine-tuned AlexNet, VGG-16 and Inception-v3 [Szegedy et al., 2015b]. For CATARACTS, Roychowdhury et al. [2017] fine-tuned Inception-v4 [Szegedy et al., 2017], ResNet-50 [He et al., 2016a] and two NASNet-A instances [Zoph et al., 2017], Hu and Heng [2017] fine-tuned ResNet-101 and DenseNet-169 [Huang et al., 2017], and Maršalkaitė et al. [2017] fine-tuned four ResNet-50 instances. Training a CNN proved challenging due to highly frequent tool co-occurrences: a solution based on label-set sampling has been proposed by Sahu et al. [2017] to reduce this bias. Note that temporal information is not exploited in these solutions, with a few exceptions presented hereafter [Sahu et al., 2017; Maršalkaitė et al., 2017; Al Hajj et al., 2017; Mishra et al., 2017; Roychowdhury et al., 2017]. In Sahu et al. [2017] and Maršalkaitė et al. [2017], a linear filter is used to smooth CNN predictions from consecutive frames. In Al Hajj et al. [2017], a CNN processes short sequences of consecutive images, using the optical flow to register and combine local features from consecutive images. In Mishra et al. [2017], one RNN processes the outputs of a frame-level CNN inside short sequences of consecutive frames. Note that long-term relationships between images are not exploited neither in these four solutions: the goal is to combine slightly different views on a tool, some of which being affected by motion blur or occlusion. In Roychowdhury et al. [2017], on the other hand, long-term relationships between images are exploited through a Markov Random Field (MRF) modeling long sequences of approximately 20,000 frames. The drawback is that online video analysis is not possible.

2.4. Proposed Solution

In this paper, we propose to design "CNN+RNN" networks, the state-of-the-art video analysis framework, for the task of automatic tool usage annotation. Due to the specific challenges of this task, namely the similarity between some tools and the rarity of some tool usages, we propose to apply the boosting principle to both the CNN part and the RNN part of the network, in a novel and unified manner. Besides addressing the previously mentioned difficult cases, the proposed framework has multiple advantages: 1) it can be used to select the network architectures automatically, an open

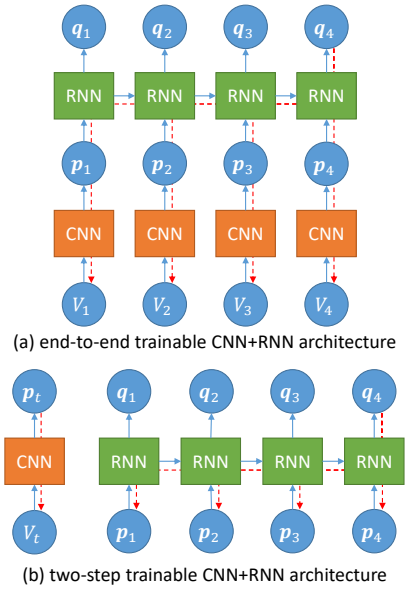


Figure 1: Training strategies for "CNN+RNN" networks. Each green cell represent one RNN cell (or several RNN cells stacked on top of each other in a multi-layer RNN). Each orange cell represents one CNN; $\boldsymbol{p}_t$ and $\boldsymbol{q}_t$ are short notations for $\boldsymbol{p}(V_t)$ and $\boldsymbol{q}(V_t)$, respectively. Two "CNN+RNN" training strategies are illustrated in Fig. (a) and (b). They reveal that the first strategy (a) is not tractable: backpropagating errors at time index t involves t backpropagations through the CNN, as illustrated in red for $t = 4$.

problem in deep learning, and 2) it can improve the complementarity of CNNs and RNNs, an unsolved problem in "CNN+RNN" models for which end-to-end learning is not tractable (see Fig. 1). Section 3 briefly describes the networks considered in this paper and the related challenges. Section 4 describes the boosting algorithm proposed to address those challenges. The proposed solution has several novelties. First, the use of CNN boosting and RNN boosting for medical images or videos is novel. Second, the data-driven design of a CNN or CNN ensemble to be used as input for an RNN or RNN ensemble (through boosting — see section 4.5) has never been studied before.

3. "CNN+RNN" Networks

3.1. Notations

Let Θ denote a set of surgical tools whose usage should be monitored in videos. Let $\mathcal{D}$ denote a collection of training videos and let V_t denote the

t-th frame in video $V \in \mathcal{D}$. Let $\delta(V_t, \theta) \in \{-1, 1\}$ denote the binary label assigned to frame V_t for tool $\theta \in \Theta$: this label indicates whether or not tool θ is being used in frame V_t. We are addressing a multilabel classification problem, so $0 \leq \sum_\theta \delta(V_t, \theta) \leq |\Theta|$. In contrast, $\sum_\theta \delta(V_t, \theta) = 1$ in a multiclass classification problem.

Neural networks considered in this paper consist of one or several CNNs working in parallel: this set of CNNs is referred to as the "CNN block". Let $\boldsymbol{p}(V_t) = \{p(V_t, \theta) \in [0; 1], \theta \in \Theta\}$ denote the instant predictions computed by the CNN block for frame V_t. Some of the neural networks considered in this paper also contain one or several RNNs working in parallel: this set of RNNs is referred to as the "RNN block". Let $\boldsymbol{q}(V_t) = \{q(V_t, \theta) \in [0; 1], \theta \in \Theta\}$ denote the context-aware predictions computed by the RNN block for frame V_t.

3.2. RNNs Processing CNN Predictions

A recurrent neural network (RNN) is a neural network that takes a sequence of observations at the input and produces a sequence of predictions at the output [Hochreiter and Schmidhuber, 1997]. In this paper, the input sequence is $\{\boldsymbol{p}(V_t)|t = 1..|V|\}$, i.e. the predictions of the CNN block for each frame in a video. The output sequence is $\{\boldsymbol{q}(V_t)|t = 1..|V|\}$. The network is structured in such a way that the prediction vector $\boldsymbol{q}(V_t)$ depends on feature vector $\boldsymbol{p}(V_t)$, but also on all previous feature vectors $\boldsymbol{q}(V_u)$, $u < t$. This behavior is achieved by 1) connecting each input element $\boldsymbol{p}(V_t)$ to a group of neurons C_t called "cell", 2) connecting C_t to the output element $\boldsymbol{q}(V_t)$ and 3) connecting C_t to the next cell C_{t+1}. Weights are shared across all cells. The most popular cells are Long Short-Term Memory (LSTM) cells [Hochreiter and Schmidhuber, 1997]: they include a "forgetting" mechanism preventing backpropagated errors from vanishing or exploding in long sequences. More recently, Gated Recurrent Units (GRU) were proposed by Cho et al. [2014]: the labeling performance of these lower-complexity cells is often comparable with LSTM.

A multi-layer extension was proposed for RNNs. In this extension, each timestamp t is associated with multiple cells $C_{i,t}$, where $i = 1..n$ is the layer index. At each timestamp t, $\boldsymbol{p}(V_t)$ is connected to $C_{1,t}$, $C_{i,t}$ is connected to $C_{i+1,t}$ for $i = 1..n-1$, and $C_{n,t}$ is connected to $\boldsymbol{p}(V_t)$. In each layer i, $C_{i,t}$ is connected to $C_{i,t+1}$. Weights are shared across all cells in the same layer. A bidirectional extension was also proposed for RNNs [Schuster and Paliwal, 1997]. In this extension, illustrated in Fig. 2, two independent RNNs are

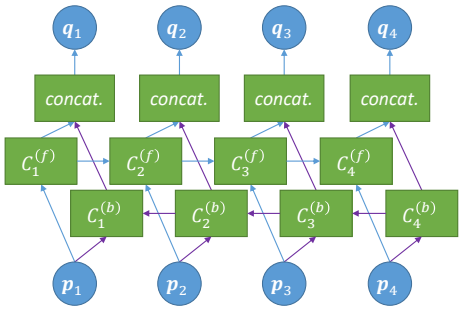


Figure 2: Bidirectional RNN networks. Three elements are defined at each timestamp: 1) a forward RNN cell (or stack of RNN cells), 2) a backward RNN cell (or stack of RNN cells) and 3) a fusion part, which concatenates their outputs. The purple arrows represent information propagated backward in time.

defined: in one of them, information flows from timestamp t to timestamp $t+1$; in the other one, information flows from timestamp t to timestamp $t-1$. Their outputs are concatenated and connected to the output sequence. The performance of bidirectional RNNs, which take advantage of past and future information, is generally higher. The drawback is of course that online video labeling is not possible.

3.3. RNNs on Long Video Sequences

In the literature, RNNs are generally trained using video sequences consisting of a few dozen frames at most [Chen et al., 2017; Gammulle et al., 2017; Mishra et al., 2017]. In contrast, analyzing all frames of full surgery videos requires the analysis of much longer sequences: for instance, there are at least 10,000 frames per video sequence in our cataract surgery videos (see section 5.1). Training long-term relationships with RNNs is more computationally intensive using long sequences, so we propose to analyze shorter sequences. In that purpose, M subsampled versions of each original sequence V, denoted by $V^{(m)}$, $m = 1..M$, are generated as follows:

$$V^{(m)} = \{V_u \mid u = m + tM, t \in \mathbb{N}^*, u \leq |V|\} \ . \tag{1}$$

During training, this results in a novel kind of data augmentation [Shen et al., 2017]: the number of training sequences increases artificially. For simplicity, $\left\{V^{(m)} \mid V \in \mathcal{D}, m = 1..M\right\}$ is denoted by $\mathcal{D}$ in the remainder of this paper. During testing, each of the M subsequences of V are analyzed independently

and the final prediction sequence for V is obtained by interleaving the resulting M prediction sequences. The resulting prediction sequence is further processed by median filters to blend subsequences: a filter of radius R_θ is used for each tool-specific channel of the sequence.

3.4. Training Complexity for "CNN+RNN" Networks

Because CNNs and RNNs are integrated into the same network, it would make sense to train the entire network from end to end, so that features extracted by the CNNs are as relevant as possible to the RNNs that process them further. However, as illustrated in Fig. 1, the complexity of the learning process is very high. The error measured for each prediction $q(V_t, \theta)$ is backpropagated to $\boldsymbol{p}(V_t)$ but also to all $\boldsymbol{p}(V_u)$ (such as $u \leq t$, in unidirectional networks). Errors computed for each $\boldsymbol{p}(V_u)$ are backpropagated further towards V_u.

The vast majority of weights in a "CNN+RNN" network are in the CNNs. Therefore, the cost of backpropagating an error measured for one timestamp t to all frames V_u in the video sequence (such as $u \leq t$, in unidirectional networks) is very high. As a consequence, a two-step training process is always preferred in the literature (see section 2.1). A CNN is trained first: errors measured for one timestamp t are only backpropagated to V_t. Then, a RNN is trained: errors measured for one timestamp t are backpropagated to all $\boldsymbol{p}(V_u)$ (such as $u \leq t$, in unidirectional networks) without affecting the CNN weights. Given the number of weights in a RNN, this process is tractable. We propose a solution based on boosting that is able to improve the CNN block after or while training the RNN block, in order to achieve the desirable properties of end-to-end training, but at a reasonable computational cost.

4. Boosted "CNN+RNN" Networks (see Fig. 3)

4.1. Context

Recent boosting algorithms, such as AnyBoost [Mason et al., 1999] and Friedman [2001]'s Gradient Boosting Machines (GBM), are formulated as a gradient descent optimization, which integrates nicely with the way neural networks are trained. When CNNs or RNNs are used as weak learners, the boosting meta-algorithm controls the loss function used to train these learners. Typically, training samples with large classification errors are assigned a larger weight in the updated loss function. A few authors thus used

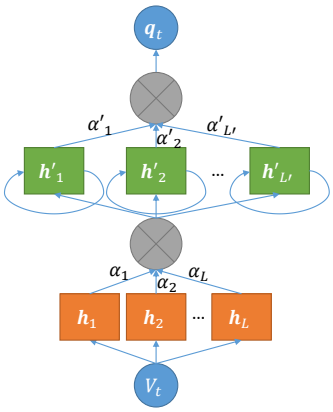


Figure 3: Boosted "CNN+RNN" network (unidirectional version). The $\otimes$ symbol represents the sigmoid operator applied to the weighted sum of the inputs.

CNNs as weak learners for AnyBoost [Moghimi et al., 2016] or GBM [Zhang et al., 2016; Walach and Wolf, 2016]. A boosting algorithm based on GBM [Friedman, 2001] is proposed in this section to design either a CNN block or an RNN block. The same algorithm is used for CNN boosting in RNN-free networks and for RNN boosting in "CNN+RNN" networks. To ensure the complementarity of the CNN and RNN blocks in "CNN+RNN" networks, an improved criterion is proposed for CNN boosting in such networks (see section 4.5). How to design an adequate neural network architecture for a given classification problem remains an open question. So, generalizing Gao et al. [2016], multiple architectures of neural networks (CNNs or RNNs) are considered in this study; let $\mathcal{H}$ denote the set of (CNN or RNN) architectures.

4.2. Gradient Boosting Machine

The purpose of GBM is to build a strong learner $\boldsymbol{H}_L$ by linearly combining multiple weak learners $\boldsymbol{h}_l \in \mathcal{H}$, l=1..L, with weights α_l. Let $\boldsymbol{h}_l(x) = \{h_l(x,\theta), \theta \in \Theta\}$ denote the predictions of $\boldsymbol{h}_l$ for some input x. The predictions of the strong learner for x are given by:

$$\boldsymbol{H}_L(x) = \sum_{l=1}^{L} \alpha_l \boldsymbol{h}_l(x) \; . \tag{2}$$

These predictions are mapped to probabilities using the sigmoid function σ: $p_L(x,\theta) = \sigma(H_L(x,\theta))$ in CNN boosting, $q_L(x,\theta) = \sigma(H_L(x,\theta))$ in RNN boosting. Weak learners are added sequentially in order to minimize the negative log-likelihood [Friedman, 2001]:

$$\mathcal{L}(\boldsymbol{h}) = -\sum_{\theta\in\Theta} \left[\sum_{x,\delta(x,\theta)=1} \log \sigma(h(x,\theta)) + \sum_{x,\delta(x,\theta)=-1} \log\left[1-\sigma(h(x,\theta))\right] \right] , \tag{3}$$

where $\delta(x,\theta)$ is the binary label assigned to x for tool θ (see section 3.1). At each boosting iteration $L+1$, all weak learners $\boldsymbol{h} \in \mathcal{H}$ are trained as detailed in sections 4.3 to 4.4. Then, the weak learner $\boldsymbol{h}$ minimizing $\mathcal{L}(\boldsymbol{H}_L + \alpha\boldsymbol{h}), \alpha \geq 0$, is added to the strong classifier:

$$(\boldsymbol{h}_{L+1}, \alpha_{L+1}) = \underset{(\boldsymbol{h}\in\mathcal{H},\alpha\geq 0)}{\operatorname{argmin}} \mathcal{L}(\boldsymbol{H}_L + \alpha\boldsymbol{h}) . \tag{4}$$

Boosting stops when $\mathcal{L}$ stops decreasing.

4.3. Loss Function for Boosting Neural Networks

As noted by Friedman [2001], the weak learner h_{L+1} selected at boosting iteration $L+1 > 1$ should ideally return values $h_{L+1}(x,\theta)$ proportional to $-\frac{\partial\mathcal{L}(\boldsymbol{H}_L)}{\partial H_L(x,\theta)}$:

$$h_{L+1}(x,\theta) = \kappa\, \omega_{L+1}(x,\theta),\ \forall x,\ \forall\theta,\ \kappa\in\mathbb{R} , \tag{5}$$

$$\omega_{L+1}(x,\theta) = -\frac{\partial\mathcal{L}(\boldsymbol{H}_L)}{\partial H_L(x,\theta)} , \tag{6}$$

where the $\omega_{L+1}(x,\theta)$ coefficients, called sample weights, are given by:

$$\omega_{L+1}(x,\theta) = \begin{cases} 1-\sigma\left(H_L(x,\theta)\right) & \text{if}\quad \delta(x,\theta)=1 \\ -\sigma\left(H_L(x,\theta)\right) & \text{if}\quad \delta(x,\theta)=-1 \end{cases} . \tag{7}$$

With that property, the strong learner's loss function would decrease directly towards zero. Neural networks can be trained to solve Eq. (5) in the least square sense, using $\kappa = 1$ without loss of generality. Therefore, the following quadratic loss function can be used for $L > 0$ [Moghimi et al., 2016]:

$$\mathcal{L}_2(\boldsymbol{h},\boldsymbol{\omega}) = \sum_{\theta}\sum_{x} \left(h(x,\theta) - \omega(x,\theta)\right)^2 . \tag{8}$$

4.4. Efficiently Training Neural Networks as Weak Learners

The proposed solution for training weak learners can be summarized as follows. At iteration 1 ($L = 0$), each weak learner $\boldsymbol{h} \in \mathcal{H}$ is trained to minimize $\mathcal{L}(\boldsymbol{h})$, the negative log likelihood [see Eq. (3)]. CNN weights are fine-tuned from a model trained on ImageNet; RNN weights are initialized at random. At iterations $L + 1$, $L > 0$, each weak learner $\boldsymbol{h} \in \mathcal{H}$ is trained to minimize $\mathcal{L}_2(\boldsymbol{h}, \boldsymbol{\omega}_{L+1})$, the quadratic loss function [see Eq. (8)]. Following Moghimi et al. [2016], the neuron weights of $\boldsymbol{h}$ are fine-tuned from neuron weights obtained at the previous boosting iteration. This strategy saves time and also improves performance. Indeed, more and more samples receive marginal weights at each boosting iteration, as the classification error decreases [see Eq. (7)]. Therefore, the training set somehow becomes smaller and smaller. The proposed strategy can be regarded as transfer learning from a larger dataset, which is known to be beneficial.

4.5. Boosting CNNs inside a "CNN+RNN" Network

The boosting solution described in previous sections is suboptimal for CNN boosting in a "CNN+RNN" network. Let us assume that one image in a video sequence is wrongly classified by the firstly selected CNN $\boldsymbol{h}_1$. Based on the temporal context, the RNN block might be able to correct this classification error. Therefore, building a second CNN $\boldsymbol{h}_2$ for correcting that error specifically might be useless. Instead, CNNs should be trained to maximize the performance of the "CNN+RNN" network as a whole.

Throughout the rest of this paper, let $\boldsymbol{H}'$, $\boldsymbol{h}'$, $\boldsymbol{\alpha}'$ and L' denote respectively the strong learner, the weak learners, their weights and their number in the RNN block, in order to avoid confusion with their counterparts in the CNN block. To achieve the desired behavior, the sample weights $\boldsymbol{\omega}_{L+1}$ should be defined based on $\boldsymbol{q}_{L'}$, the outputs of the RNN block, rather than $\boldsymbol{p}_L$, the outputs of the CNN block: the goal should be to minimize $\mathcal{L}(\boldsymbol{H}_L, \boldsymbol{H'}_{L'})$. In this scenario, $\omega_{L+1}(V_t, \theta)$, the weight assigned to frame V_t and label $\theta \in \Theta$, does not depend solely on instant quantities, namely $\boldsymbol{H}_L(V_t)$ and $\delta(V_t, \theta)$. In bidirectional networks (for offline processing), it depends on all $(\boldsymbol{H}_L(V_u), \delta(V_u, \phi))$ pairs, $\phi \in \Theta$. In unidirectional networks, it depends

on all pairs such that $u \geq t$. For $L > 0$, sample weights become:

$$\begin{cases} \omega_{L+1}(V_t, \theta) &= p_L(V_t, \theta)(1 - p_L(V_t, \theta)) \\ &\times \sum_{\phi \in \Theta} \sum_{V_u} \Delta^{\delta(V_u, \phi)}(V_t, \theta, V_u, \phi) \\ \Delta^{+}(V_t, \theta, V_u, \phi) &= (1 - q_{L'}(V_u, \phi)) \sum_{l=1}^{L'} \alpha'_l \frac{\partial h'_l(V_u, \phi)}{\partial p_L(V_t, \theta)} \\ \Delta^{-}(V_t, \theta, V_u, \phi) &= -q_{L'}(V_u, \phi) \sum_{l=1}^{L'} \alpha'_l \frac{\partial h'_l(V_u, \phi)}{\partial p_L(V_t, \theta)} \end{cases} . \tag{9}$$

If a unidirectional RNN network is used, then the $\partial h'_l(V_u, \phi)/\partial p_L(V_t, \theta)$ partial derivatives equal zero for all $u < t$. In all other cases, they can be computed automatically by the backpropagation algorithm. Note that the backpropagation algorithm does not compute each $\frac{\partial O_i}{\partial I_j}$ term individually, where I denotes an input tensor whose influence on the output tensor O should be computed. Instead, it computes:

$$\sum_i \frac{\partial O_i}{\partial I_j} \nabla_i, \tag{10}$$

given a tensor ∇ weighting each coefficient of the output tensor. However, Eq. (9) can be computed setting:

- $O_i = h'_l(V_u, \phi)$, $i = (u, \phi)$,
- $I_j = p_L(V_t, \theta)$, $j = (t, \theta)$,
- $\nabla_i = 1 - q_{L'}(V_u, \phi)$ or $\nabla_i = q_{L'}(V_u, \phi)$ depending on $\Delta^{\delta(V_u, \phi)}$.

Proof for Eq. (9). In this scenario, the partial derivative of the negative log-likelihood function [see Eq. (3)], with respect to $H_L(V_t, \theta)$, is given by:

$$\frac{\partial \mathcal{L}(\boldsymbol{H}_L, \boldsymbol{H'}_{L'})}{\partial H_L(V_t, \theta)} = -\sum_{\phi \in \Theta} \left[\sum_{V_u, \delta(V_u, \phi) = 1} \frac{\partial \log q_{L'}(V_u, \phi)}{\partial H_L(V_t, \theta)} + \sum_{V_u, \delta(V_u, \phi) = -1} \frac{\partial \log \left(1 - q_{L'}(V_u, \phi)\right)}{\partial H_L(V_t, \theta)} \right] . \tag{11}$$

Each term in this sum can be decomposed according to the chain rule of derivation, using the following equations:

$$\frac{\partial \log \sigma(y)}{\partial \sigma(y)} = \frac{1}{\sigma(y)} , \tag{12}$$

$$\frac{\partial \log (1-\sigma(y))}{\partial \sigma(y)} = \frac{-1}{1-\sigma(y)} , \tag{13}$$

$$\frac{\partial q_{L'}(V_u,\phi)}{\partial H_L(V_t,\theta)} = \frac{\partial q_{L'}(V_u,\phi)}{\partial \sigma(H_L(V_t,\theta))} \frac{\partial \sigma(H_L(V_t,\theta))}{\partial H_L(V_t,\theta)} . \tag{14}$$

The second factor on the right hand side of Eq. (14) can be decomposed using the derivative of the sigmoid function:

$$\frac{\partial \sigma(y)}{\partial y} = \sigma(y)(1-\sigma(y)) , \tag{15}$$

Similarly, the first factor on the right hand side of Eq. (14) can be decomposed as follows:

$$\frac{\partial q_{L'}(V_u,\phi)}{\partial p_L(V_t,\theta)} = q_{L'}(V_u,\phi)(1-q_{L'}(V_u,\phi)) \sum_{l=1}^{L'} \alpha'_l \frac{\partial h'_l(V_u,\phi)}{\partial p_L(V_t,\theta)} . \tag{16}$$

where $q_{L'}(V_u,\phi) = \sigma(H_{L'}(V_u,\phi))$ and $H_{L'}(V_u,\phi)$ is a function of all $p_L(V_t,\theta)$ values.

The sample weights we have defined for CNN boosting inside a "CNN+RNN" network are more complex than the general case [see Eq. 7]. However, they are only computed once per boosting iteration. Therefore, they do not make the optimization problem significantly less tractable, as opposed to the end-to-end training of a "CNN+RNN" network. But, like end-to-end training, they ensure a good complementarity between the CNN and RNN blocks.

4.6. Joint CNN and RNN Boosting

Two strategies are proposed below to define the order in which CNNs and RNNs are trained to design data-driven "CNN+RNN" architectures.

"Sequential" strategy. The most straightforward solution is to boost the CNN block while $\mathcal{L}(\boldsymbol{H}_L)$ decreases, and then to boost the RNN block while $\mathcal{L}(\boldsymbol{H'}_{L'})$ decreases. Besides the use of boosting, this is the standard approach for designing "CNN+RNN" networks (see section 2.1). However, this solution suffers from the limitation described in the previous section, namely the lack of complementarity between the CNN and RNN blocks.

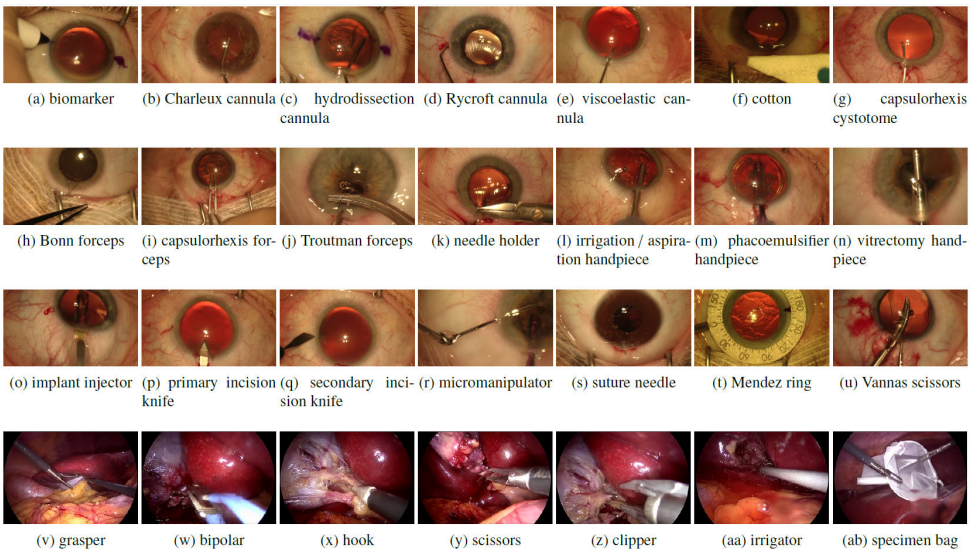


Figure 4: Surgical tools annotated in videos

"Joint" strategy. To overcome this limitation, we propose to design the CNN and RNN blocks inside a single boosting loop, using a single strong learner's loss function, namely $\mathcal{L}(\boldsymbol{H}_L, \boldsymbol{H'}_{L'})$. At each boosting iteration, all CNN architectures $\boldsymbol{h} \in \mathcal{H}$ and all RNN architectures $\boldsymbol{h'} \in \mathcal{H}'$ are trained (or re-trained) and only one CNN or one RNN is added to the network: the one minimizing

$$\begin{array}{l} \{\mathcal{L}(\boldsymbol{H}_L + \alpha\boldsymbol{h}, \boldsymbol{H'}_{L'}) \mid \boldsymbol{h} \in \mathcal{H}, \alpha \geq 0\} \\ \bigcup \{\mathcal{L}(\boldsymbol{H}_L, \boldsymbol{H'}_{L'} + \alpha'\boldsymbol{h'}) \mid \boldsymbol{h'} \in \mathcal{H}', \alpha' \geq 0\} \end{array} . \tag{17}$$

Of course, in the first boosting iteration, only CNN architectures are considered: RNNs need at least one feature extractor to operate. Eq. (9) is used to define the sample weights for CNN boosting as soon as $L' \geq 1$.

5. Surgery Video Datasets

The proposed approach is applied to tool usage annotation in two surgical video datasets: CATARACTS and Cholec80.

Dataset	Tool	Inter-rater agreement	% of training frames
CATARACTS	biomarker	0.835	0.0168 %
	Charleux cannula	0.963	1.79 %
	hydrodissection cannula	0.982	2.43 %
	Rycroft cannula	0.919	3.18 %
	viscoelastic cannula	0.975	2.54 %
	cotton	0.947	0.751 %
	capsulorhexis cystotome	0.995	4.42 %
	Bonn forceps	0.798	1.10 %
	capsulorhexis forceps	0.849	1.62 %
	Troutman forceps	0.764	0.258 %
	needle holder	0.630	0.0817 %
	irrigation/aspiration handpiece	0.995	14.2%
	phacoemulsifier handpiece	0.997	15.3 %
	vitrectomy handpiece	0.998	2.76 %
	implant injector	0.980	1.41 %
	primary incision knife	0.961	0.700 %
	secondary incision knife	0.852	0.522 %
	micromanipulator	0.995	17.6 %
	suture needle	0.893	0.219 %
	Mendez ring	0.953	0.100 %
	Vannas scissors	0.823	0.0443 %
Cholec80	grasper	n/a	55.3 %
	bipolar	n/a	4.47 %
	hook	n/a	56.7 %
	scissors	n/a	1.76 %
	clipper	n/a	3.29 %
	irrigator	n/a	5.05 %
	specimen bag	n/a	6.35 %

Table 1: Statistics about tool usage annotation in the CATARACTS and Cholec80 datasets. The first column indicates inter-rater agreement (Cohen's kappa) after adjudication. The last column indicates the prevalence of each tool in the training set (excluding frames without a consensus in CATARACTS).

5.1. CATARACTS Dataset

The CATARACTS dataset contains 50 videos of cataract surgeries performed in Brest University Hospital.[5] The purpose of cataract surgeries is

[5]https://cataracts.grand-challenge.org

to remove a clouded natural lens and replace it with an artificial lens. The entire procedure can be performed with small incisions only. Surgeries were monitored through an OPMI Lumera T microscope (Carl Zeiss Meditec, Jena, Germany). Videos were recorded with a 180I camera (Toshiba, Tokyo, Japan) and a MediCap USB200 recorder (MediCapture, Plymouth Meeting, USA). The frame definition was 1920x1080 pixels and the frame rate was approximately 30 frames per second (fps). Videos had a duration of 10 minutes and 56 s on average (minimum: 6 minutes 23 s, maximum: 40 minutes 34 s). In total, more than nine hours of surgery have been video recorded. A list of 21 tools visible in these videos was compiled by a surgeon (see Fig 4). Then, the usage of each tool in videos was annotated independently by two non-clinical experts, after an initial training by a surgeon. A tool was considered to be in use whenever it was in contact with the eyeball. Therefore, both experts recorded a timestamp whenever one tool started or stopped touching the eyeball. Tool-tissue contacts can be detected well: they imply deformations of the eye surface, which are well visible thanks to specular reflections of light. Finally, annotations from both experts were adjudicated: whenever experts disagreed about the label of one tool, they watched the video together and jointly determined the actual label. However, the precise timing of tool/eyeball contacts was not adjudicated. Inter-rater agreement after adjudication is reported in Table 1. The dataset was divided into a training set (25 videos) and a test set (25 videos). Division was made in such a way that each tool appears in the same number of videos from both subsets (plus or minus one). The classification performance for θ was assessed only in frames where experts agreed about the usage of θ. During training, some tool $\theta \in \Theta$ was considered to be in use if at least one expert said so.

5.2. Cholec80 Dataset

The Cholec80 dataset contains 80 videos of cholecystectomy surgeries [Twinanda et al., 2017]. The purpose of cholecystectomy is to remove the gallbladder: this operation can be performed laparoscopically and monitored through an endoscope. Videos were recorded with a frame definition of 1920x1080 pixels and a frame rate of 25 fps. Videos had a duration of 38 minutes and 26 s on average (minimum: 12 minutes 19 s, maximum: 1 hour 39 minutes 55 s). They were downsampled to 1 fps for processing. In total, more than 51 hours of surgery have been video recorded (2 hours after downsampling). In Cholec80, a tool was considered to be in use if it was visible through the endoscope (if at least half of the tool tip was visible, precisely).

The presence of seven tools was annotated in videos (see Fig 4): one binary label is provided per image and per tool. The dataset was divided into a training set (40 videos) and a test set (40 videos).

5.3. Training and Validation Subsets

For validation purposes, two training videos of CATARACTS (respectively four videos of Cholec80) were assigned to a validation subset; the remaining training videos were assigned to a learning subset used to optimize the CNN, RNN and boosting weights. In CATARACTS, the validation videos were chosen such that all tools appear in the learning subset: it was not possible to ensure this property for both subsets. In Cholec80, they were chosen at random.

6. Experiments

6.1. Architectures

Seven CNN architectures were used as weak classifiers in this paper:

- VGG-16 and VGG-19 [Simonyan and Zisserman, 2015],
- the second version [He et al., 2016b] of ResNet-101 and ResNet-152 [He et al., 2016a],
- Inception-v4 and Inception-ResNet-v2 [Szegedy et al., 2017],
- NASNet-A [Zoph et al., 2017].

The TensorFlow-Slim implementation[6] of these CNNs was used, with weights pre-trained on ImageNet. The last layer of each CNN, which computes one logit prediction per class, was resized from 1000 neurons for ImageNet to 21 neurons for CATARACTS or 7 neurons for Cholec80; the weights of these neurons were initialized at random. The same input image size was used for ImageNet, CATARACTS and Cholec80: 224×224 pixels for VGG-16 and VGG-19, 299×299 pixels for ResNet-101, ResNet-152, Inception-v4 and Inception-ResNet-v2, and 331×331 pixels for NASNet-A. To preserve the aspect ratio, images from CATARACTS and Cholec80 were first resized to 224×126 pixels, 299×168 pixels or 331×184 pixels and were then padded

[6] `https://github.com/tensorflow/models/tree/master/research/slim`

CNN	single image	batch processing
VGG-16	7.50 ms / image	2.87 ms / image
VGG-19	8.50 ms / image	3.44 ms / image
ResNet-101	10.2 ms / image	3.16 ms / image
ResNet-152	13.2 ms / image	4.62 ms / image
Inception-v4	18.8 ms / image	6.09 ms / image
Inception-ResNet-v2	19.0 ms / image	6.34 ms / image
NASNet-A	24.6 ms / image	18.5 ms / image

Table 2: Inference times of CNNs using one GeForce GTX 1080 Ti GPU by Nvidia. Inference times are given for batch processing (mini-batches of 16 images for NASNet-A and 32 images for other CNNs), which can be used for offline video labeling, and for single image processing, which must be used for online video labeling.

with zeros at the top and the bottom to obtain square images. All CNNs were trained using the RMSProp algorithm with a learning rate initialized to 0.01 and decaying exponentially. In order to define a more challenging boosting problem, we conducted a secondary experiment involving the three worst performing CNNs only: this experiment is called "weaker CNNs", while the primary experiment involving all CNNs is called "all CNNs".

Regarding RNN boosting, two types of RNN cells were used: LSTM [Hochreiter and Schmidhuber, 1997] and GRU [Cho et al., 2014]. To limit complexity and computation times, the number of layers in RNNs was set to $n = 2$. Three different values were used for C, the number of neurons per cell, in order to define six weak classifiers (three based on LSTM, three based on GRU): $C = 64$, $C = 128$, $C = 256$. In all RNN boosting experiments, a subsampling factor of $M = 16$ and $M = 4$ was used in CATARACTS and Cholec80, respectively: this number was found to be optimal in initial experiments on the validation subset (see Fig. 5). All RNNs were trained using the RMSProp algorithm with a constant learning rate of 0.001. As for the median filter radii R_θ, they were selected within $\{1, 2, 4, 8, 16, 32, 64\}$ to maximize the classification performance in the validation set; for rare tools absent from the validation set, the most frequently selected value was used. RNNs were implemented using Keras version 2.0.8.

Inference times for CNNs, the most computationally intensive parts of the system, are reported in Table 2.

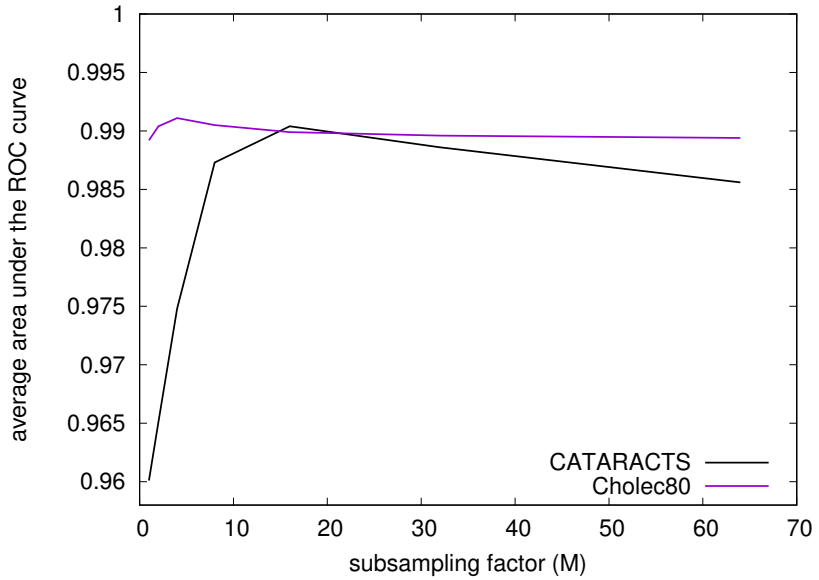


Figure 5: Effect of the subsampling factor M (which is also the data augmentation rate — see section 3.3) on tool annotation performance in the validation subset. This figure reports the average performance obtained using NASNet-A and each of the six weak RNN classifiers based on LSTM or GRU.

6.2. Performance of Boosted Video Labelers

The performance of the seven weak CNNs is reported in Tables 3 and 4. As expected, the best performing CNN, NASNet-A, is also the most recent. Surprisingly, VGG-19 and VGG-16 are also quite good, in spite of being older and less sophisticated than the others. The three worst performing CNNs (in the validation set and in the test set) are ResNet-101, ResNet-152 and Inception-ResNet-v2: they were used in the "weaker CNNs" experiment. The architecture of boosted bidirectional video labelers are reported in Fig 6 for the "all CNNs" and "weaker CNNs" experiments. Their performance is detailed in Tables 3 and 4 for the "all CNNs" experiment. In the largest dataset (CATARACTS), training the initial CNNs with early stopping took between 2h (ResNet-101) and 11h (Inception-ResNet-v2); training NASNet-A took 8h. In the following boosting iterations, fine-tuning the CNNs and training/fine-tuning the RNNs took 3h at most per CNN or RNN. At each boosting iterations, CNNs and RNNs were trained in parallel on a cluster of GeForce GTX 1080 Ti GPUs (RNNs were trained without GPU). Over-

all, if the process was fully-automated, boosting would have lasted approximately 29h. In practice, it took a few days, as the process involved manual interactions (for early stopping in particular). The end-to-end training of a NASNet-A + RNN network would have lasted more than 80,000 hours (9 years) for CATARACTS, which involves sequences of more than 10,000 frames.

Tables 3 and 4 show that "CNN+RNN" boosting improves performance compared to CNN boosting alone in both datasets. Median filtering also improves performance in the CATARACTS dataset but decreases it in Cholec80. For each tool θ, the least worst radius is $R_\theta = 1$ for Cholec80 and the best radius is $2 \leq R_\theta \leq 32$ for CATARACTS. ROC curves and precision-recall curves for the best CNN, namely NASNet-A, and the best ensemble, namely joint "CNN+RNN" boosting (with median filtering for CATARACTS), are reported in Fig. 7 and 8. In terms of area under the ROC curve (A_z), all tools were detected well by the best ensemble ($A_z \geq 0.9694$). In terms of average precision (AP), rare tools are poorly detected before boosting ($AP < 0.1$ in some cases). For rare tools, precision (and therefore AP) is indeed impacted strongly by the number of false alarms which, in the specificity criterion (and therefore A_z), is divided by the large number of negative samples. In fact, as shown in Table 4, AP is highly correlated with tool prevalence in the training set. However, the mean AP is greatly improved after boosting: from mAP = 0.6086 to mAP = 0.7980 in CATARACTS. Since there are no rare tools in Cholec80, mAP is much higher (up to mAP = 0.9789).

Sequence labeling examples obtained with the best ensembles are illustrated and commented in Fig. 9. In summary, mistakes made by the best ensembles are mainly due to occlusions. To illustrate the problems that the proposed ensemble solves, Fig. 10 reports labeling sequences obtained at different ensemble complexity levels. This figure suggests that the same errors are made by all detectors, but these errors are progressively attenuated as the ensemble becomes more complex.

6.3. Comparisons with Baseline Solutions

The proposed ensemble (obtained through "CNN+RNN boosting", with median filtering for CATARACTS) is compared with various baseline methods in Table 5. For each baseline, the statistical significance of the difference with the proposed solution is assessed using a paired sample t-test.

The first five baselines are variations on the proposed ensemble, as described above. All of these variations lead to decreased performance, with

Dataset	Tool	VGG-16	VGG-19	ResNet-101	ResNet-152	Inception-v4	Inception-ResNet-v2	NASNet-A	boosted CNN	boosted CNN+RNN	smoothed boosted CNN+RNN
CATARACTS	biomarker	0.9364	0.9855	0.9469	*0.9948*	0.7852	0.6313	0.9544	0.9920	0.9997	**0.9998**
	Charleux cannula	0.9105	0.9360	0.8238	0.8813	0.9129	0.9174	*0.9603*	0.9677	0.9877	**0.9903**
	hydrodissection cannula	0.9807	0.9875	0.9585	0.9709	*0.9911*	0.9768	0.9807	0.9938	0.9956	**0.9965**
	Rycroft cannula	0.9858	0.9846	0.9747	0.9776	0.9857	0.9743	*0.9896*	0.9920	0.9948	**0.9952**
	viscoelastic cannula	0.9441	0.9341	0.9347	0.8709	0.9393	0.9224	*0.9628*	0.9630	0.9670	**0.9711**
	cotton	0.9879	0.9889	0.9417	0.9848	0.9622	0.9472	*0.9910*	0.9926	0.9966	**0.9973**
	capsulorhexis cystotome	0.9935	0.9978	0.9950	0.9955	0.9981	0.9940	*0.9982*	0.9990	0.9997	**0.9998**
	Bonn forceps	0.9769	*0.9896*	0.9745	0.9813	0.9867	0.9768	0.9860	0.9933	0.9945	**0.9949**
	capsulorhexis forceps	0.9706	0.9767	0.9648	0.9641	*0.9896*	0.9765	0.9878	0.9944	0.9983	**0.9986**
	Troutman forceps	0.9746	0.9811	0.9790	0.9472	0.9844	0.9766	*0.9886*	0.9901	0.9969	**0.9971**
	needle holder	0.9667	*0.9911*	0.9329	0.9312	0.9722	0.9762	0.9908	0.9951	0.9974	**0.9981**
	irrigation/aspiration HP	0.9950	0.9960	0.9879	0.9910	0.9961	0.9913	*0.9963*	0.9981	0.9991	**0.9993**
	phacoemulsifier HP	0.9969	0.9983	0.9939	0.9968	0.9980	0.9969	*0.9984*	0.9992	**0.9998**	**0.9998**
	vitrectomy HP	0.9756	0.9761	0.9874	0.9516	0.9812	*0.9888*	0.9579	0.9930	0.9894	**0.9927**
	implant injector	0.9811	0.9827	0.9790	0.9772	*0.9887*	0.9797	0.9765	0.9909	0.9943	**0.9952**
	primary incision knife	0.9881	0.9908	0.9819	0.9686	*0.9959*	0.9809	0.9814	0.9969	0.9994	**0.9996**
	secondary incision knife	0.9924	0.9976	0.9977	*0.9989*	0.9982	0.9980	0.9972	0.9989	0.9996	**0.9997**
	micromanipulator	0.9919	0.9943	0.9919	0.9913	*0.9959*	0.9922	0.9953	0.9972	0.9983	**0.9985**
	suture needle	0.9757	*0.9779*	0.9504	0.9742	0.9647	0.9612	0.9756	0.9844	0.9989	**0.9991**
	Mendez ring	0.9943	*0.9997*	0.9939	0.9792	0.9435	0.9724	0.9913	0.9970	**1.0000**	**1.0000**
	Vannas scissors	*0.9939*	0.9924	0.9810	0.9648	0.9799	0.9662	0.9842	0.9943	0.9947	**0.9961**
	Average (mA_z)	0.9768	*0.9837*	0.9653	0.9663	0.9690	0.9570	0.9831	0.9916	0.9953	**0.9961**
	Corr. with prevalence	0.3246	0.2305	0.2793	0.2850	0.2955	0.2461	0.3922	0.2493	0.1309	0.1220
Cholec80	grasper	*0.9633*	0.9620	0.9472	0.9539	0.9505	0.9523	0.9618	0.9689	**0.9694**	0.9652
	bipolar	*0.9949*	0.9946	0.9929	0.9904	0.9903	0.9913	0.9941	0.9962	**0.9977**	0.9975
	hook	0.9983	0.9983	0.9972	0.9975	0.9963	0.9973	*0.9984*	0.9989	**0.9991**	0.9966
	scissors	0.9877	0.9865	0.9771	0.9751	0.9802	0.9819	*0.9903*	0.9909	0.9958	**0.9959**
	clipper	0.9977	0.9979	0.9955	0.9958	0.9923	0.9954	*0.9983*	0.9989	**0.9998**	**0.9998**
	irrigator	0.9932	*0.9935*	0.9859	0.9895	0.9861	0.9882	0.9930	0.9956	**0.9980**	0.9975
	specimen bag	*0.9951*	*0.9951*	0.9915	0.9914	0.9926	0.9937	0.9944	0.9969	0.9976	**0.9977**
	Average (mA_z)	*0.9900*	0.9897	0.9839	0.9848	0.9840	0.9857	*0.9900*	0.9923	**0.9939**	0.9929
	Corr. with prevalence	-0.4916	-0.4865	-0.4317	-0.3751	-0.4396	-0.4555	-0.5149	-0.5186	-0.5915	-0.6535

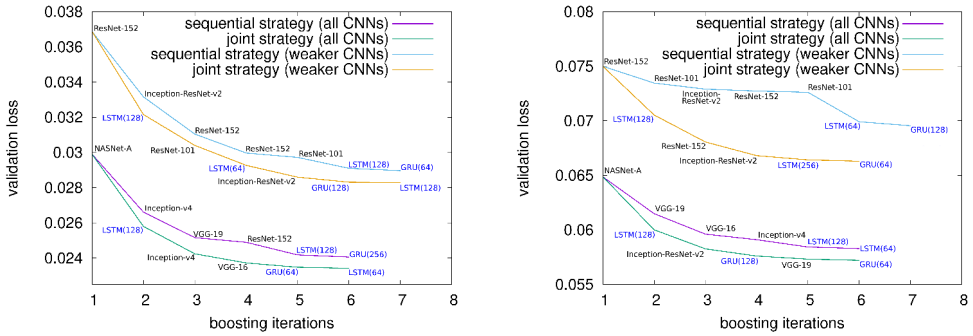
Table 3: Areas under the ROC curves (A_z) for each weak CNN classifier and strong classifiers in the "all CNNs" experiment. In case of "CNN+RNN" boosting, the "joint" strategy is used. HP stands for "handpiece". On each line, the highest score is marked in bold and the highest score among weak CNN classifiers is marked in italic. For each dataset, the last row indicates the Pearson correlation between A_z in the test set and tool prevalence in the training set (see Table 1).

Dataset	Tool	VGG-16	VGG-19	ResNet-101	ResNet-152	Inception-v4	Inception-ResNet-v2	NASNet-A	boosted CNN	boosted CNN+RNN	smoothed boosted CNN+RNN
CATARACTS	biomarker	0.0046	0.0120	0.0039	0.0482	0.0012	0.0005	*0.1294*	0.1311	0.5628	**0.6352**
	Charleux cannula	0.0538	0.0891	0.0473	0.1353	0.1276	0.1594	*0.4386*	0.2455	0.5728	**0.6003**
	hydrodissection cannula	0.8339	0.8652	0.7870	0.8211	*0.8881*	0.8141	0.8678	0.9213	0.9412	**0.9471**
	Rycroft cannula	0.7819	0.7095	0.7381	0.7357	0.8085	*0.7807*	*0.8530*	0.8637	0.9084	**0.9155**
	viscoelastic cannula	0.5670	0.6065	0.5925	0.5582	0.6178	0.4828	*0.7048*	0.6833	0.7588	**0.7658**
	cotton	0.0063	0.0071	0.0101	0.1317	0.1093	*0.2491*	0.1092	0.1474	0.2308	**0.3148**
	capsulorhexis cystotome	0.9356	0.9700	0.9399	0.9510	0.9768	0.9462	*0.9786*	0.9868	0.9959	**0.9968**
	Bonn forceps	0.6181	0.7007	0.6580	0.7102	*0.7251*	0.5806	0.4816	0.7805	0.8174	**0.8223**
	capsulorhexis forceps	0.6319	0.6441	0.6343	0.6399	*0.7705*	0.6600	0.6956	0.8210	0.8950	**0.9023**
	Troutman forceps	0.2468	0.2408	0.3803	0.2779	0.3714	0.3282	*0.4200*	0.4348	0.6173	**0.6474**
	needle holder	0.1371	0.2420	0.0495	0.0514	0.0709	0.1586	*0.2916*	0.2504	0.5197	**0.6356**
	irrigation/aspiration HP	0.9818	0.9848	0.9551	0.9652	*0.9854*	0.9765	0.9846	0.9919	0.9954	**0.9964**
	phacoemulsifier HP	0.9877	0.9940	0.9813	0.9904	0.9923	0.9889	*0.9949*	0.9967	0.9991	**0.9992**
	vitrectomy HP	0.4175	0.3869	0.6402	0.5496	0.4219	***0.7296***	0.4154	0.6454	0.6088	0.6430
	implant injector	0.7701	0.7836	0.8331	0.8400	*0.8693*	0.8026	0.8524	0.8965	0.9353	**0.9386**
	primary incision knife	0.7944	0.8644	0.8121	0.8443	*0.9081*	0.7628	0.7823	0.9203	0.9696	**0.9740**
	secondary incision knife	0.6524	0.8321	0.8434	*0.9195*	0.9120	0.7946	0.8903	0.9169	0.9615	**0.9649**
	micromanipulator	0.9777	0.9843	0.9773	0.9786	*0.9878*	0.9767	0.9867	0.9920	0.9950	**0.9955**
	suture needle	0.3740	*0.4728*	0.3937	0.3957	0.4006	0.2942	0.3983	0.4702	0.8031	**0.8204**
	Mendez ring	0.1439	*0.8266*	0.0759	0.0220	0.0083	0.0587	0.4292	0.3696	0.9606	**0.9977**
	Vannas scissors	*0.1987*	0.1723	0.1284	0.0441	0.0930	0.0713	0.0760	0.2127	0.1937	**0.2456**
	Average (mAP)	0.5293	0.5899	0.5467	0.5529	0.5736	0.5532	*0.6086*	0.6513	0.7734	**0.7980**
	Corr. with prevalence	0.6474	0.5560	0.5974	0.5918	0.5600	0.6382	0.6166	0.5544	0.4443	0.4329
Cholec80	grasper	*0.9723*	0.9711	0.9621	0.9656	0.9627	0.9646	0.9711	0.9764	**0.9767**	0.9730
	bipolar	0.9667	0.9643	0.9513	0.9491	0.9452	0.9477	*0.9688*	0.9740	**0.9823**	0.9781
	hook	*0.9986*	*0.9986*	0.9977	0.9980	0.9971	0.9979	*0.9986*	0.9990	**0.9992**	0.9961
	scissors	0.8802	0.8810	0.8101	0.8241	0.8120	0.8072	*0.8993*	0.9155	0.9465	**0.9501**
	clipper	0.9729	0.9722	0.9490	0.9567	0.9326	0.9446	*0.9814*	0.9860	**0.9958**	0.9952
	irrigator	0.9568	*0.9572*	0.9213	0.9330	0.9178	0.9319	0.9561	0.9692	**0.9781**	0.9657
	specimen bag	0.9555	0.9551	0.9310	0.9365	0.9364	0.9453	*0.9605*	0.9691	**0.9735**	0.9729
	Average (mAP)	0.9576	0.9571	0.9318	0.9376	0.9291	0.9342	*0.9623*	0.9699	**0.9789**	0.9759
	Corr. with prevalence	0.5477	0.5529	0.5899	0.5867	0.6371	0.5741	0.5261	0.4964	0.3838	0.3845

Table 4: Average precision (AP) for each weak CNN classifier and strong classifiers in the "all CNNs" experiment. In case of "CNN+RNN" boosting, the "joint" strategy is used. HP stands for "handpiece". On each line, the highest score is marked in bold and the highest score among weak CNN classifiers is marked in italic. For each dataset, the last row indicates the Pearson correlation between AP in the test set and tool prevalence in the training set (see Table 1).

	CATARACTS		Cholec80		p-value (paired sample t-test)	
	mA_z	mAP	mA_z	mAP	difference in mA_z	difference in mAP
proposed ensemble	0.9961	0.7980	0.9939	0.9789		
boosted CNN	0.9916	0.6513	0.9923	0.9699	2.964×10^{-4}	0.001789
boosted CNN (weaker CNNs only)	0.9829	0.6192	0.9880	0.9501	2.479×10^{-4}	0.003881
proposed ensemble (weaker CNNs only)	0.9900	0.6748	0.9917	0.9695	0.007271	0.01401
smoothed "sequentially" boosted CNN+RNN	0.9939	0.6956	0.9930	0.9741	0.002679	0.004047
proposed ensemble (unidirectional RNNs)	0.9957	0.7580	0.9936	0.9760	**0.05397**	**0.07474**
NASNet-A	0.9831	0.6086	0.9900	0.9623	1.836×10^{-5}	5.321×10^{-5}
NASNet-A + 1 LSTM	0.9900	0.6949	0.9911	0.9723	0.001088	0.001972
NASNet-A features + 1 LSTM	0.9910	0.7264	0.9913	0.9755	0.02051	0.01078
boosted CNN + smoothing	0.9933	0.6735	0.9917	0.9703	0.001317	0.003009
linear-combination CNN ensemble	0.9913	0.6611	0.9917	0.9674	5.328×10^{-4}	0.001204
smoothed linear-combination CNN+LSTM ensemble	0.9937	0.7010	0.9926	0.9733	0.002235	0.004602
EndoNet [Twinanda et al., 2017]	n/a		n/a	0.810	n/a	0.005394
DResSys [Roychowdhury et al., 2017]	0.9971	n/a	n/a		**0.2936**	n/a
CUMV [Hu and Heng, 2017]	0.9897	n/a	n/a		0.001682	n/a
TROLIS [Maršalkaitė et al., 2017]	0.9812	n/a	n/a		0.0204	n/a
proposed ensemble (union GT)	0.9938	0.7876	n/a		0.02266	**0.2320**
proposed ensemble (intersection GT)	0.9958	0.7585			0.001920	0.007045

Table 5: Comparisons between the proposed ensemble (jointly boosted "CNN+RNN" ensemble, with median filtering for CATARACTS) and various baselines, in terms of mean area under the ROC curve (mA_z) and in terms of mean average precision (mAP). Non-significant differences at the 95% confidence level are in bold. All CNNs are used to build ensembles unless specified otherwise (the weaker CNNs are Inception-ResNet-v2, ResNet-101 and ResNet-152). The last two experiments evaluate the proposed ensemble using the union or the intersection of tool usage annotations from both experts as ground truth.

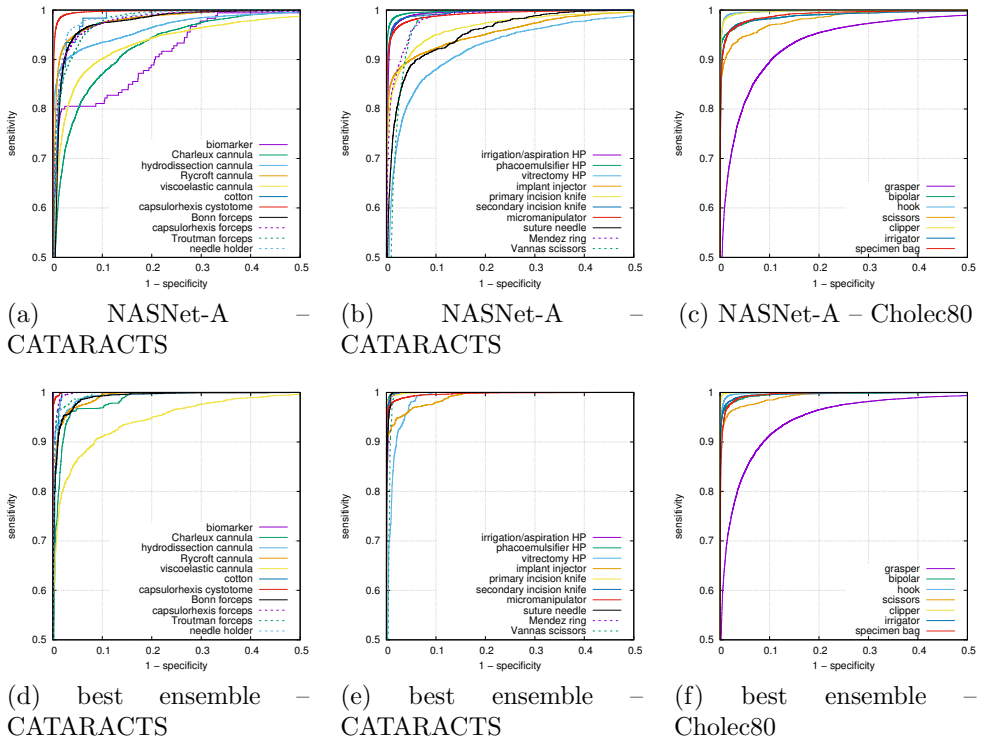


(a) CATARACTS (b) Cholec80

Figure 6: Evolution of the validation loss across boosting iterations, using bidirectional RNNs. The number of neurons in RNN cells is indicated in brackets. Curves and architectures obtained for the unidirectional version are very similar: they are not reported.

one exception: replacing bidirectional RNNs with unidirectional RNNs does not impact performance significantly. We note the good performance of ensembles obtained in the "weaker CNNs" experiment. In CATARACTS for instance, A_z increases from 0.9663 for the best CNN (ResNet-152) to 0.9900 (+0.0237), while in the "all CNNs" experiment, it increases from 0.9831 to 0.9961 (+0.0130). On the downside, we also note that to achieve very high performance, good weak learners must be available.

The next six baselines were proposed to evaluate the relevance of each part of the proposed framework. The first two tests show that only using the best CNN (NASNet-A in the "all CNNs" experiment) or the best CNN and the best RNN (NASNet-A + 1 LSTM) is clearly suboptimal. Interestingly, the third experiment shows that LSTMs operating on NASNet-A features (the 4032 features of the next to last NASNet-A layer) are better than LSTMs operating on NASNet-A predictions (the outputs of the last layer). However, besides being more computationally intensive, RNNs operating on CNN features are not compatible with boosting across multiple CNN architectures: feature layers would not necessarily have compatible shapes and could therefore not be combined linearly. The fourth experiment shows that the RNN part of the proposed ensemble cannot simply be replaced with a median filter. The ensemble evaluated in the fifth experiment is similar to the proposed boosted CNN ensemble, in the sense that predictions from

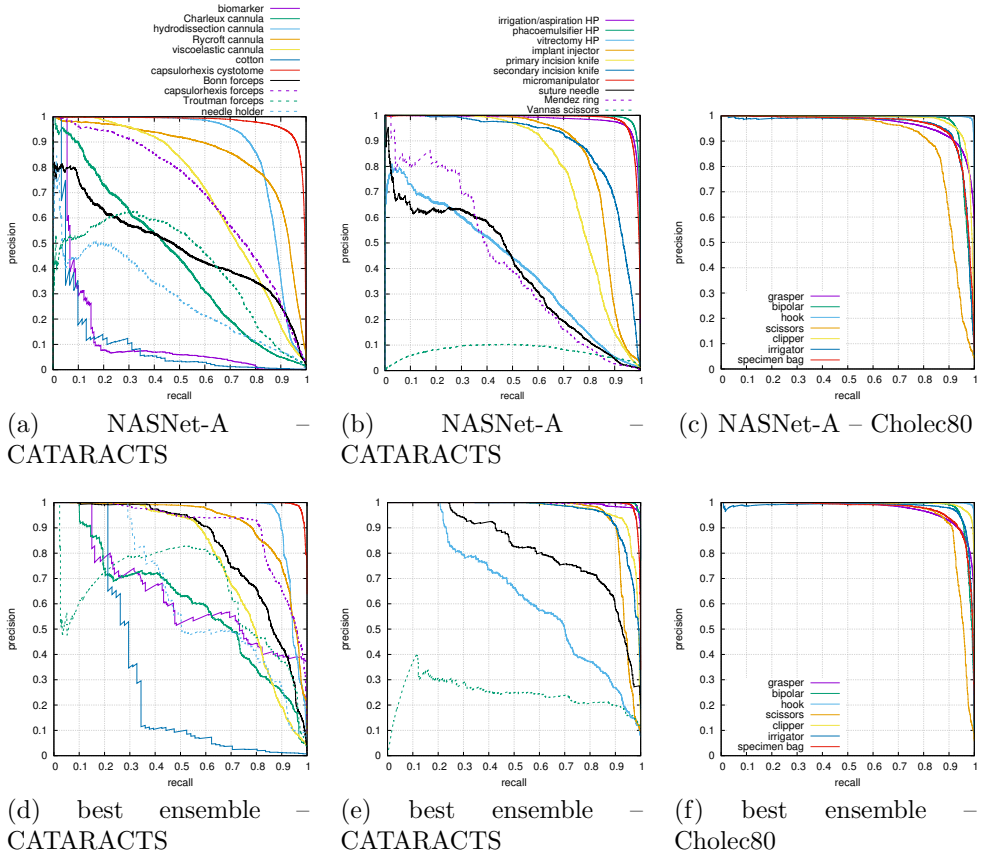


(a) NASNet-A – CATARACTS (b) NASNet-A – CATARACTS (c) NASNet-A – Cholec80

(d) best ensemble – CATARACTS (e) best ensemble – CATARACTS (f) best ensemble – Cholec80

Figure 7: Receiver-operating characteristic (ROC) curves for the best weak classifier (NASNet-A) and the best ensemble of the "all CNNs" experiment (jointly boosted "CNN+RNN" architecture, with median filtering for CATARACTS). Note that only the top left quadrant of the ROC space (sensitivity and specificity ≥ 0.5) is displayed for improved visualization.

several CNNs are combined linearly inside a sigmoid function. The difference is that each CNN (the seven CNNs studied in this paper) is trained independently; the weight assigned to each CNN is trained through a gradient descent. This approach is similar to the ensemble method proposed by Roychowdhury et al. [2017]. The result of this experiment is rather disappointing: the performance of the resulting ensemble is almost as good as the boosted CNN ensemble ($p = 0.2390$ for A_z, $p = 0.7066$ for AP). The only advantage of the proposed ensemble is that it is more compact: four CNNs (see Fig. 6) instead of seven. A similar ensemble is evaluated in the sixth experiment: one LSTM is trained independently on top of each of the seven

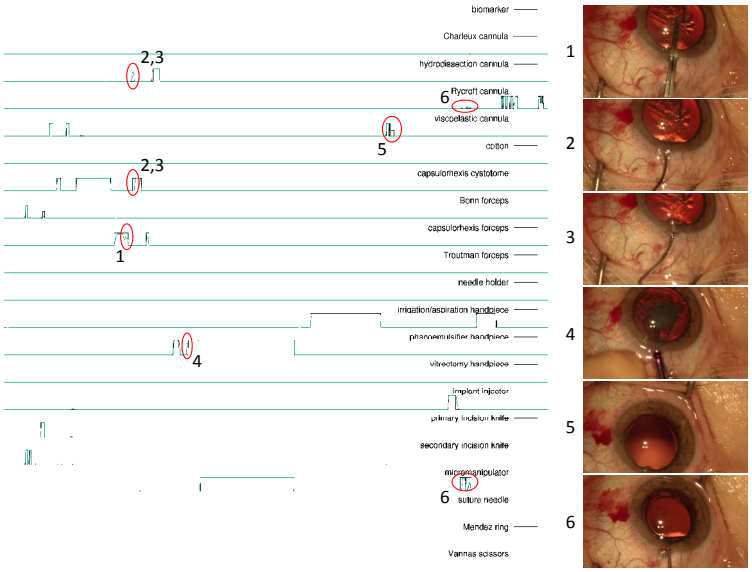


(a) NASNet-A – CATARACTS (b) NASNet-A – CATARACTS (c) NASNet-A – Cholec80

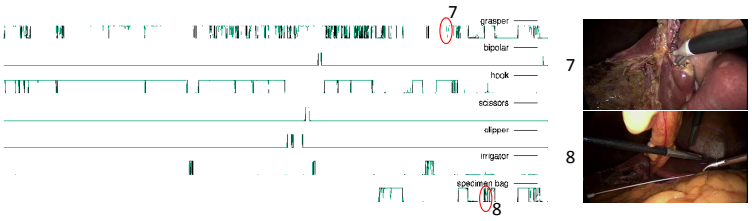
(d) best ensemble – CATARACTS (e) best ensemble – CATARACTS (f) best ensemble – Cholec80

Figure 8: Precision-recall (PR) curves for the best weak classifier (NASNet-A) and the best ensemble of the "all CNNs" experiment (jointly boosted "CNN+RNN" architecture, with median filtering for CATARACTS). Fig. (a) and (d) share the same legend. The same applies to Fig. (b) and (e).

CNNs and the predictions of these seven LSTMs is combined linearly inside a sigmoid function, again with weights obtained through a gradient descent. In CATARACTS, the ensemble predictions are then smoothed with a median filter. In that case, the proposed boosting approach is superior. We assume this superiority is mainly due to the proposed mechanism for boosting CNNs inside a "CNN+RNN" network (see section 4.5), since the performance of the linear-combination ensemble is close to that of the "sequentially" boosted "CNN+RNN". Ideally, we would also compare the proposed solution with the



(a) test video from CATARACTS



(b) test video from Cholec80

Figure 9: Sequence labeling for one test video from each dataset using the best ensemble of the "all CNNs" experiment (joint "CNN+RNN" Boosting, with median filtering for CATARACTS): tool usage according to human experts is in black, automatic predictions are in green. Areas surrounded by red circles are associated with images on the right. The label of image 1 (capsulorhexis forceps) has been correctly identified, but with a lower confidence level compared to previous images. The reason probably is that the forceps have remained closed for a long time and are therefore more difficult to recognize. In image 2, the capsulorhexis cystotome is detected as a hydrodissection cannula. The reason probably is that its distinctive claw-shaped tooltip is hidden in the incision and its distinctive elbow is out of the field of view. As soon as the elbow becomes visible (image 3), the correct label is assigned. In image 4, the phacoemulsifier handpiece is considered active, whereas it is not in contact with the eyeball yet. However, it touches the tear film, so the detector is almost correct. In image 5, one of the annotators indicated that the viscoelastic cannula is being used, although it is not actually visible: only indirect signs of presence (at the bottom) are visible; the detector was not able to recognize them. In image 6, the micromanipulator is partly mistaken for a Rycroft cannula: the explanation is similar for images 2 and 6. The reason why a hydrodissection cannula is detected in the former case and a Rycroft cannula in the latter probably comes from the RNN-based temporal modeling: hydrodissection cannulae are more likely at the beginning, Rycroft cannula are more likely at the end. In image 7, the grasper is not detected, probably because it is occluded by the hook. Finally, in image 8, a specimen bag is falsely detected, however the white string used for closing the bag is visible: the RNN-based temporal sequencer probably interpolated predictions from neighboring frames where the bag and the string are both visible.

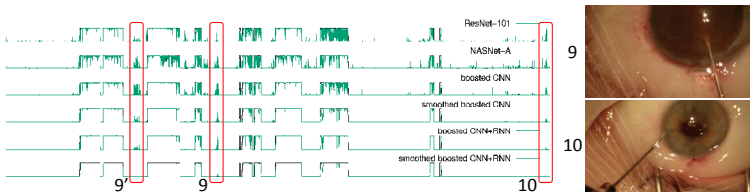


Figure 10: Sequence labeling for the micromanipulator tool for one test video from CATARACTS: tool usage according to human experts is in black, automatic predictions are in green. Areas surrounded by red circles are associated with images on the right. In images 9 and 9', the viscoelastic cannula is falsely detected as a micromanipulator. In image 10, a Rycroft cannula is falsely detected as a micromanipulator.

end-to-end training of a "CNN+RNN" network, but the complexity of that model prevents any experimentation.

The next four baselines are recent solutions from the literature: EndoNet is from the original Cholec80 paper, the other three solutions are the top-ranking solutions of the CATARACTS challenge. We can see that the proposed solution is better than three of these solutions (EndoNet, CUMV and TROLIS) and not significantly worse than the other one (DResSys). One advantage of the proposed solution compared to DResSys is that it is more lightweight (less CNNs processing smaller images). The other advantage is that its unidirectional version, which is not significantly different from DResSys neither ($p = 0.07525$), allows online video sequencing, while DResSys jointly analyzes batches of ∼20,000 frames.

The last two experiments reported in Table 5 evaluate the impact of the criterion chosen to define the ground truth in CATARACTS (exclusion of frames without a consensus). In those experiments, the ground truth is defined either as the union or the intersection of both expert interpretations, using all frames in the test videos. We can see that using those evaluation criteria decreases performance, in part because the most challenging frames (where experts disagree) are included, in part because the ground truth is of lower quality (more uncertain).

6.4. Sensitivity Analysis of the Boosted Video Labelers

To visualize what the CNNs have learned, one can rely on sensitivity analysis [Simonyan et al., 2014] and related metrics. Sensitivity is the gradient of the CNN predictions with respect to the pixel values: the pixel values influencing most the CNN predictions are highlighted. Recently, we proposed

a variation on sensitivity called hue-constrained sensitivity [Quellec et al., 2017]: the interpretation is similar, except that the three color components of a pixel are analyzed jointly rather than independently. Given a CNN $\boldsymbol{h}$ and an input image I with dimensions $W \times H \times 3$, the hue-constrained sensitivity heatmap π of I for $\boldsymbol{h}$ is defined as:

$$\pi_{x,y} = \left| \frac{\partial \sum_{\theta \in \Theta} h\left(m * I, \theta\right)}{\partial m_{x,y}} \right| \quad , \tag{18}$$

where tensor m is a matrix of ones with dimensions $W \times H$ and where '*' denotes the element-wise tensor multiplication. It should be noted that $m * I = I$ and that all color components of a pixel in I are multiplied by the same tensor element in m, which ensures the desired hue preservation property [Quellec et al., 2017]. Fig. 11 reports hue-constrained sensitivity heatmaps for all seven CNNs. It also reports heatmaps for $\boldsymbol{h}_2$, the second CNN (based on Inception-v4) added to the strong classifier in the "all CNNs" experiment (where $\boldsymbol{h}_1$ is NASNet-A). This figure shows that, in CATARACTS, CNNs do not consider solely the tools, but also the anterior segment of the eye: the lens, which is modified by tools, the cornea, which is temporarily deformed by tools as they move, and the corneoscleral junction, where tools are inserted. One explanation is that each tool interacts differently with the eye and, therefore, analyzing the eye structures helps differentiating tools. Another explanation is that, in this dataset, the target labels are not related to tool presence, but rather to tool usage. So CNNs must be able to recognize whenever each tool is in contact with the eye. This hypothesis is backed up by the observation that responses from tissues are lower in Cholec80, where tool usage is simply defined as tool visibility. We notice, however, that the best CNNs (NASNet-A and VGG-19) have sparser heatmaps and that those heatmaps are more focused on the tools. Heatmaps obtained for $\boldsymbol{h}_1$ (i.e NASNet-A) and $\boldsymbol{h}_2$ have been analyzed jointly to assess their complementarity. Because the first image was already classified well by NASNet-A, the heatmap for $\boldsymbol{h}_2$ is empty: the detections we see at the corner are just amplified noise (heatmap intensities have been normalized between 0 and 255). Similarly, in the second image, the phacoemulsifier handpiece at the center was detected well by NASNet-A, but not the forceps on the left: $\boldsymbol{h}_2$ seems to focus on the forceps. In the third image, we note that NASNet-A did not focus primarily on the tool (it seemed disturbed by specular reflections) but $\boldsymbol{h}_2$ does. In the last image, we also note a more focused heatmap for $\boldsymbol{h}_2$, compared to NASNet-A, although the grasper on the left (which was

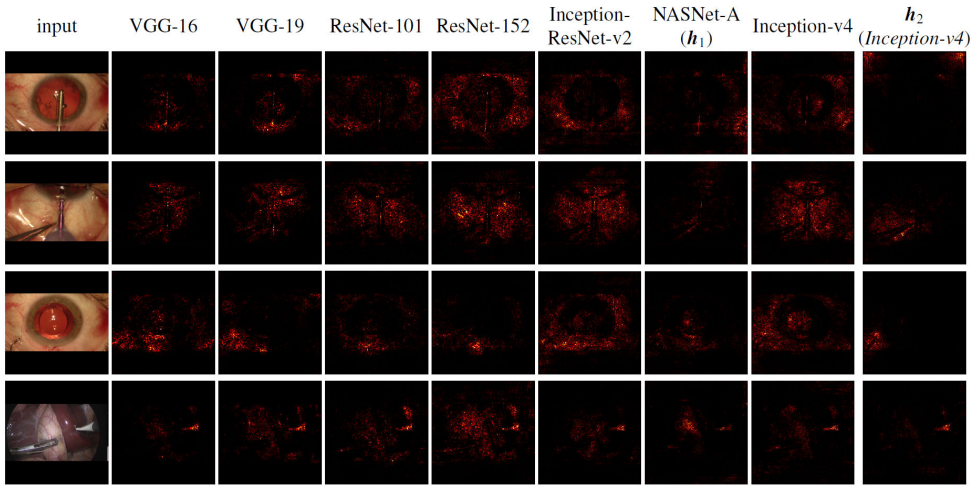


Figure 11: Hue-constrained sensitivity analysis for multiple CNNs. The first three examples were taken from the the test set of CATARACTS. The last example was taken from the the test set of Cholec80. $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are the first two CNNs selected in the "all CNNs" experiment.

correctly detected by $\boldsymbol{h}_1$) is not detected anymore. So, overall, $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are indeed complementary. And, clearly, the heatmaps for Inception-v4 before and after a boosting step are very different.

Because our joint "CNN+RNN" boosting algorithm relies on the gradients of RNN predictions with respect to CNN predictions [see Eq. (9)], sensitivity analysis is also useful for RNNs in our case. These gradients are illustrated in a condensed form in Fig. 12: given an RNN $\boldsymbol{h}'$, this figure shows $\nabla_{\phi,\theta}(\boldsymbol{h}')$, where:

$$\nabla_{\phi,\theta}(\boldsymbol{h}') = \sum_{\boldsymbol{V}\in\mathcal{D}} \sum_{t} \sum_{u} \frac{\partial h'(V_u,\phi)}{\partial p_L(V_t,\theta)} \,. \tag{19}$$

For a lazy RNN, all coefficients outside the diagonal would be zero. Here, we observe that the diagonal is not even always dominant. This is particularly true for tools whose detection performance increases greatly after RNN boosting, such as needle holders, suture needles or Cholec80's scissors (see Tables 3 and 4): the gradients of RNN predictions for those tools with respect to CNN predictions for other tools are very high. Clearly, RNNs are not lazy and quite useful for this task.

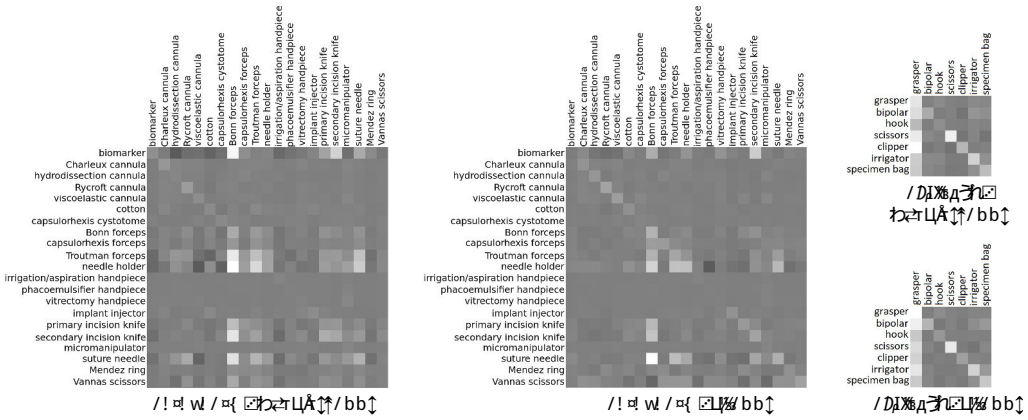


Figure 12: Sensitivity analysis for $\boldsymbol{h}'_1$, the first added RNN in the two experiments based on joint "CNN+RNN" boosting: the "all CNNs" or "3 weakest CNNs" experiments. Intensity is proportional to $\nabla_{\phi,\theta}(\boldsymbol{h}'_1)$ [see Eq. (19)]: gray means zero, black means negative, white means positive. Rows represent ϕ, the label index in RNN predictions. Columns represent θ, the label index in CNN predictions.

7. Discussion and Conclusions

A solution for labeling tool usage in cataract and cholecystectomy surgery videos has been presented. Following state-of-the-art video analysis solutions, it relies on convolutional neural networks (CNNs) for analyzing each frame in the video and on recurrent neural networks (RNNs) for analyzing the temporal sequencing throughout the entire surgery, based on the outputs of the CNNs. A novel framework for boosting a sequence labeler composed of CNNs and RNNs has been presented. The main motivation for this framework is the fact that "CNN+RNN" labelers cannot be trained from end to end, for complexity reasons. The framework allows to progressively improve the CNN and RNN parts of the system by adding weak classifiers (CNNs or RNNs) designed to improve the overall classification accuracy of the join system. In particular, like the theoretical end-to-end training solution, CNN training is supervised based on the outputs of the RNN block.

The proposed framework has several novelties. The main novelty lies in the boosting algorithm. CNN boosting had been proposed for multiclass classification problems [Moghimi et al., 2016]. We adapted it for multilabel classification, showed its applicability to RNN boosting and, more importantly, introduced CNN boosting supervised based on the outputs of the

RNN block. A second novelty lies in the proposed temporal sequence augmentation strategy: although very simple, it proved to be quite effective (see Fig. 5).

The proposed framework is quite general and is likely applicable outside the scope of surgery video analysis. However, it is of particular relevance for this application because many tools are very similar to one another (e.g. the cannulae or the forceps — see Fig. 4) but they are often used in a predefined order: using the temporal context (e.g. which tools have be used previously) is quite relevant for differentiating them. Therefore, it seems particularly useful to guide CNN training or boosting based on the temporal context. Experiments on two recent datasets (CATARACTS and Cholec80) for the task of tool usage annotation demonstrated its very good performance: the mean area under the ROC curve reaches up to m$A_z = 0.9961$ over a collection of 21 cataract surgery tools and up to m$A_z = 0.9939$ over a collection of 7 cholecystectomy tools.

If we look into the details of the proposed boosting solution, we first note that CNN boosting alone is disappointing: we found no significant difference between CNN boosting and a weighted sum of independently trained CNNs ($p = 0.2390$ for A_z, $p = 0.7066$ for AP), although the resulting architecture is more lightweight. The ability to boost CNNs based on the outputs of RNNs, on the other hand, leads to a significant improvement: joint "CNN+RNN" boosting is indeed significantly better than sequential "CNN+RNN" boosting ($p = 0.002679$ for A_z, $p = 0.004047$ for AP — see Table 5). Our explanation is that, when the CNN part is boosted independently of RNNs, much boosting effort is spent on trying to correct labeling errors, caused by previously selected CNNs, that RNNs could easily correct based on the temporal context: using temporally-filtered outputs to supervise boosting makes more sense. These observations support our hypothesis that CNNs should be trained to be complimentary to RNNs.

One advantage of the proposed approach is that its online version, which relies on unidirectional RNNs, does not perform significantly worse than its offline version, relying on bidirectional RNNs ($p = 0.05397$ for A_z, $p = 0.07474$ for AP — see Table 5). With slightly better performance, the offline version would be the preferred solution for report generation, surgical workflow optimization and surgical skill assessment. The online version, however, is the only valid solution for intraoperative warning or recommendation generation, provided that it is fast enough. Similarly to the bidirectional version (see Fig. 6), the online version relies on three weak CNNs: one based on

NASNet-A, one based on Inception-v4 and one based on VGG-16. All three together, processing one frame takes 50.9 ms using one GeForce GTX 1080 Ti GPU by Nvidia (see Table 2). Videos of the CATARACTS dataset have a frame rate of 30 image per second (i.e. 33.3 ms per image). It means a faster GPU would be required for real-time video analysis. Alternatively, two GPUs can be used, as the CNN classifiers can be run in parallel (GPU 1: NASNet-A $\rightarrow$ 24.6 ms per image, GPU 2: Inception-v4 and VGG-16 $\rightarrow$ 26.3 ms per image). Note that the use of median filters (with radii of 32 frames at most) delays predictions by one second. In Cholec80, the frame rate is 1 image per second, so computation times are not an issue.

The proposed framework compares favorably with state-of-the-art competing solutions [Twinanda et al., 2017; Hu and Heng, 2017; Maršalkaitė et al., 2017]. In terms of A_z, it does not differ significantly from the winner of the CATARACTS challenge [Roychowdhury et al., 2017]. However, it has the advantage of being more lightweight and, more importantly, of allowing online video analysis.

This study has a few limitations. In particular, the same dataset was used to train CNNs and RNNs. Because CNN predictions are likely better in the learning set than in the validation and test sets, RNNs are trained under too favorable conditions, which could lead to overfitting. Because the number of learning videos is limited, we decided to use all of them for training CNNs and RNNs. We simply relied on early stopping to discard overfitted configurations. Another limitation is that we did not explore data rebalancing techniques [Sahu et al., 2017] or weighted cost functions to deal with multi-label imbalance, assuming that boosting can deal with it satisfactorily.

In conclusion, an accurate solution for labeling tool usage in surgery videos has been presented. In view of the good performance, automatic surgery monitoring can now be envisaged seriously [Charrière et al., 2017]. We are currently exploring solutions to provide useful feedbacks to the surgeon, based on information collected during the surgery. Support to beginners is a particular relevant application, but many more can be envisioned for the near future.

References

Al Hajj, H., Lamard, M., Charrière, K., Cochener, B., and Quellec, G. (2017). Surgical tool detection in cataract surgery videos through multi-image fu-

sion inside a convolutional neural network. In *Proc IEEE EMBC*, Jeju Island, Korea.

Bodenstedt, S., Wagner, M., Katić, D., Mietkowski, P., Mayer, B., Kenngott, H., Müller-Stich, B., Dillmann, R., and Speidel, S. (2017). Unsupervised temporal context learning using convolutional neural networks for laparoscopic workflow analysis. Technical Report arXiv:1702.03684 [cs], Karlsruhe Institute of Technology.

Bouget, D., Allan, M., Stoyanov, D., and Jannin, P. (2017). Vision-based and marker-less surgical tool detection and tracking: a review of the literature. *Med Image Anal*, 35:633–654.

Cadène, R., Robert, T., Thome, N., and Cord, M. (2016). M2cai workflow challenge: convolutional neural networks with time smoothing and hidden Markov model for video frames classification. Technical Report arXiv:1610.05541 [cs], Université de Pierre et Marie Curie.

Charrière, K., Quellec, G., Lamard, M., Martiano, D., Cazuguel, G., Coatrieux, G., and Cochener, B. (2017). Real-time analysis of cataract surgery videos using statistical models. *Multimed Tools Appl*, 76(21):22473–22491.

Chen, H., Chen, J., Hu, R., Chen, C., and Wang, Z. (2017). Action recognition with temporal scale-invariant deep learning framework. *China Communications*, 14(2):163–172.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. In *Proc SSST*, pages 103–111, Doha, Qatar. arXiv: 1409.1259.

Dergachyova, O., Bouget, D., Huaulmé, A., Morandi, X., and Jannin, P. (2016). Automatic data-driven real-time segmentation and recognition of surgical workflow. *Int J Comput Assist Radiol Surg*, 11(6):1081–1089.

Dipietro, R., Lea, C., Malpani, A., Ahmidi, N., Vedula, S., Lee, G., Lee, M., and Hager, G. (2016). Recognizing surgical activities with recurrent neural networks. In *Proc MICCAI*, pages 551–558, Athens, Greece.

Donahue, J., Hendricks, L., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., and Darrell, T. (2017). Long-term recurrent convolutional

networks for visual recognition and description. *IEEE Trans Pattern Anal Mach Intell*, 39(4):677–691.

Feng, Y., Li, Y., and Luo, J. (2016). Learning effective gait features using LSTM. In *Proc IEEE ICPR*, pages 325–330, Cancun, Mexico.

Freund, Y. and Schapire, R. E. (1997). A Decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci*, 55(1):119–139.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann Stat*, 29(5):1189–1232.

Gammulle, H., Denman, S., Sridharan, S., and Fookes, C. (2017). Two stream LSTM: A deep fusion framework for human action recognition. In *Proc IEEE WACV*, pages 177–186, Santa Rosa, CA, USA.

Gao, Y., Rong, W., Shen, Y., and Xiong, Z. (2016). Convolutional neural network based sentiment analysis using Adaboost combination. In *Proc IEEE IJCNN*, pages 1333–1338, Vancouver, Canada.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proc CVPR*, pages 770–778, Las Vegas, NV, USA.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *Proc ECCV*, Lecture Notes in Computer Science, pages 630–645, Amsterdam, The Netherlands. Springer, Cham.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput*, 9(8):1735–1780.

Hu, X. and Heng, P.-A. (2017). Surgical tool annotation in cataract surgery videos. Technical report, Chinese University of Hong Kong.

Huang, G., Liu, Z., Maaten, L. v. d., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proc IEEE CVPR*, pages 2261–2269, Honolulu, HI, USA.

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Trans Pattern Mach Intell*, 35(1):221–231.

Jin, Y., Dou, Q., Chen, H., Yu, L., and Heng, P.-A. (2016). EndoRCN: recurrent convolutional networks for recognition of surgical workflow in cholecystectomy procedure video. Technical report, The Chinese University of Hong Kong.

Khorrami, P., Le, P., Brady, K., Dagli, C., and Huang, T. (2016). How deep neural networks can improve emotion recognition on video data. In *Proc IEEE ICIP*, pages 619–623, Phoenix, AZ, USA.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Proc NIPS*, volume 25, pages 1097–1105, Granada, Spain.

Lalys, F. and Jannin, P. (2014). Surgical process modelling: a review. *Int J Comput Assist Radiol Surg*, 9(3):495–511.

Lea, C., Vidal, R., and Hager, G. D. (2016a). Learning convolutional action primitives for fine-grained action recognition. In *Proc IEEE ICRA*, pages 1642–1649, Stockholm, Sweden.

Lea, C., Vidal, R., Reiter, A., and Hager, G. (2016b). Temporal convolutional networks: a unified approach to action segmentation. In *Proc ECCV*, pages 47–54, Amsterdam, The Netherlands.

Maršalkaitė, G., Bialopetravičius, J., and Armaitis, J. (2017). Towards robust tool identification for cataract surgery. Technical report, Oxipit, UAB.

Mason, L., Baxter, J., Bartlett, P. L., and Frean, M. R. (1999). Boosting algorithms as gradient descent. In *Proc NIPS*, volume 12, pages 512–518, Denver, CO, USA.

Mishra, K., Sathish, R., and Sheet, D. (2017). Learning latent temporal connectionism of deep residual visual abstractions for identifying surgical tools in laparoscopy procedures. In *Proc IEEE CVPR Works*, pages 2233–2240, Honolulu, HI, USA.

Moghimi, M., Saberian, M., Yang, J., Li, L.-J., Vasconcelos, N., and Belongie, S. (2016). Boosted convolutional neural networks. In *Proc BMVC*, York, UK.

Primus, M., Putzgruber-Adamitsch, D., Taschwer, M., Münzer, B., El-Shabrawi, Y., Böszörmenyi, L., and Schoeffmann, K. (2018). Frame-based classification of operation phases in cataract surgery videos. In *Proc MMM*, volume 10704 LNCS, pages 241–253, Bangkok, Thailand.

Quellec, G., Charrière, K., Boudi, Y., Cochener, B., and Lamard, M. (2017). Deep image mining for diabetic retinopathy screening. *Med Image Anal*, 39:178–193.

Quellec, G., Lamard, M., Cochener, B., and Cazuguel, G. (2014). Real-time segmentation and recognition of surgical tasks in cataract surgery videos. *IEEE Trans Med Imaging*, 33(12):2352–2360.

Quellec, G., Lamard, M., Cochener, B., and Cazuguel, G. (2015). Real-time task recognition in cataract surgery videos using adaptive spatiotemporal polynomials. *IEEE Trans Med Imaging*, 34(4):877–887.

Raju, A., Wang, S., and Huang, J. (2016). M2CAI surgical tool detection challenge report. Technical report, University of Texas at Arlington.

Roychowdhury, S., Bian, Z., Vahdat, A., and William G., M. (2017). Identification of surgical tools using deep neural networks. Technical report, D-Wave Systems Inc.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *Int J Comput Vis*, 115(3):211–252.

Sahu, M., Mukhopadhyay, A., Szengel, A., and Zachow, S. (2016). Tool and phase recognition using contextual CNN features. Technical Report arXiv:1610.08854 [cs.CV], Zuse Institute Berlin.

Sahu, M., Mukhopadhyay, A., Szengel, A., and Zachow, S. (2017). Addressing multi-label imbalance problem of surgical tool detection using CNN. *Int J Comput Assist Radiol Surg*, 12(6):1013–1020.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Trans Signal Process*, 45(11):2673–2681.

Shen, D., Wu, G., and Suk, H.-I. (2017). Deep learning in medical image analysis. *Annu Rev Biomed Eng*, 19:221–248.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: visualising image classification models and saliency maps. In *ICLR Workshop*, Calgary, Canada.

Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Proc NIPS*, volume 27, pages 568–576, Montreal, Canada.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proc ICLR*, San Diego, CA, USA.

Singh, B., Marks, T., Jones, M., Tuzel, O., and Shao, M. (2016). A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proc IEEE CVPR*, pages 1961–1970, Las Vegas, NV, USA.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proc AAAI*, pages 4278–4284, San Francisco, CA, USA.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015a). Going deeper with convolutions. In *Proc IEEE CVPR*, pages 1–9, Boston, MA, USA.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015b). Rethinking the Inception architecture for computer vision. Technical Report arXiv:1512.00567 [cs], Google.

Tao, L., Zappella, L., Hager, G. D., and Vidal, R. (2013). Surgical gesture segmentation and recognition. In *Proc MICCAI*, pages 339–346, Nagoya, Japan.

Tran, D., Sakurai, R., Yamazoe, H., and Lee, J.-H. (2017). Phase segmentation methods for an automatic surgical workflow analysis. *Int J Biomed Imaging*, 2017:1985796.

Trikha, S., Turnbull, A. M. J., Morris, R. J., Anderson, D. F., and Hossain, P. (2013). The journey to femtosecond laser-assisted cataract surgery: new beginnings or a false dawn? *Eye (Lond)*, 27(4):461–473.

Twinanda, A. P., Mutter, D., Marescaux, J., de Mathelin, M., and Padoy, N. (2016). Single- and multi-task architectures for surgical workflow challenge at M2cai 2016. Technical Report arXiv:1610.08844 [cs], University of Strasbourg.

Twinanda, A. P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M., and Padoy, N. (2017). EndoNet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Trans Med Imaging*, 36(1):86–97.

Walach, E. and Wolf, L. (2016). Learning to count with CNN boosting. In *Proc ECCV*, volume 9906, pages 660–676, Amsterdam, The Netherlands.

Wang, X., Gao, L., Song, J., and Shen, H. (2017). Beyond frame-level CNN: saliency-aware 3D CNN with LSTM for video action recognition. *IEEE Signal Processing Letters*, 24(4):510–514.

Zappella, L., Béjar, B., Hager, G., and Vidal, R. (2013). Surgical gesture classification from video and kinematic data. *Med Image Anal*, 17(7):732–745.

Zhang, F., Du, B., and Zhang, L. (2016). Scene classification via a gradient boosting random convolutional network framework. *IEEE Trans Geosci Remote Sens*, 54(3):1793–1802.

Zia, A., Castro, D., and Essa, I. (2016). Fine-tuning deep architectures for surgical tool detection. Technical report, Georgia Institute of Technology.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2017). Learning transferable architectures for scalable image recognition. *arXiv:1707.07012 [cs, stat]*.

CATARACTS: Challenge on Automatic Tool Annotation for cataRACT Surgery

Hassan Al Hajj[a], Mathieu Lamard[b,a], Pierre-Henri Conze[c,a], Soumali Roychowdhury[d], Xiaowei Hu[e], Gabija Maršalkaitė[f], Odysseas Zisimopoulos[g], Muneer Ahmad Dedmari[i], Fenqiang Zhao[k], Jonas Prellberg[l], Manish Sahu[m], Adrian Galdran[p], Teresa Araújo[o,p], Duc My Vo[q], Chandan Panda[r], Navdeep Dahiya[s], Satoshi Kondo[t], Zhengbing Bian[d], Arash Vahdat[d], Jonas Bialopetravičius[f], Evangello Flouty[g], Chenhui Qiu[k], Sabrina Dill[m], Anirban Mukhopadhyay[n], Pedro Costa[p], Guilherme Aresta[o,p], Senthil Ramamurthy[s], Sang-Woong Lee[q], Aurélio Campilho[o,p], Stefan Zachow[m], Shunren Xia[k], Sailesh Conjeti[i,j], Danail Stoyanov[g,h], Jogundas Armaitis[f], Pheng-Ann Heng[e], William G. Macready[d], Béatrice Cochener[b,a,u], Gwenolé Quellec[a,*]

[a] *Inserm, UMR 1101, Brest, F-29200 France*
[b] *Univ Bretagne Occidentale, Brest, F-29200 France*
[c] *IMT Atlantique, LaTIM UMR 1101, UBL, Brest, F-29200 France*
[d] *D-Wave Systems Inc., Burnaby, BC, V5G 4M9 Canada*
[e] *Dept. of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China*
[f] *Oxipit, UAB, Vilnius, LT-10224 Lithuania*
[g] *Digital Surgery Ltd, EC1V 2QY, London, UK*
[h] *University College London, Gower Street, WC1E 6BT, London, UK*
[i] *Chair for Computer Aided Medical Procedures, Faculty of Informatics, Technical University of Munich, Garching b. Munich, 85748 Germany*
[j] *German Center for Neurodegenrative Diseases (DZNE), Bonn, 53127 Germany*
[k] *Key Laboratory of Biomedical Engineering of Ministry of Education, Zhejiang University, HangZhou, 310000 China*
[l] *Dept. of Informatics, Carl von Ossietzky University, Oldenburg, 26129 Germany*
[m] *Department of Visual Data Analysis, Zuse Institute Berlin, Berlin, 14195 Germany*
[n] *Department of Computer Science, Technische Universität Darmstadt, 64283 Darmstadt, Germany*
[o] *Faculdade de Engenharia, Universidade do Porto, Porto, 4200-465 Portugal*
[p] *INESC TEC - Instituto de Engenharia de Sistemas e Computadores - Tecnologia e Ciência, Porto, 4200-465 Portugal*

*LaTIM - IBRBS - CHRU Morvan - 12, Av. Foch
29609 Brest CEDEX - FRANCE
Tel.: +33 2 98 01 81 29 / Fax: +33 2 98 01 81 24
Email address: gwenole.quellec@inserm.fr (Gwenolé Quellec)

[q]Gachon University, 1342 Seongnamdaero, Sujeonggu, Seongnam 13120, Korea
[r]Epsilon, Bengaluru, Karnataka 560045, India
[s]Laboratory of Computational Computer Vision, Georgia Tech, Atlanta, GA 30332, USA
[t]Konica Minolta, Inc., Osaka, 569-8503 Japan
[u]Service d'Ophtalmologie, CHRU Brest, Brest, F-29200 France

Abstract

Surgical tool detection is attracting increasing attention from the medical image analysis community. The goal generally is not to precisely locate tools in images, but rather to indicate which tools are being used by the surgeon at each instant. The main motivation for annotating tool usage is to design efficient solutions for surgical workflow analysis, with potential applications in report generation, surgical training and even real-time decision support. Most existing tool annotation algorithms focus on laparoscopic surgeries. However, with 19 million interventions per year, the most common surgical procedure in the world is cataract surgery. The CATARACTS challenge was organized in 2017 to evaluate tool annotation algorithms in the specific context of cataract surgery. It relies on more than nine hours of videos, from 50 cataract surgeries, in which the presence of 21 surgical tools was manually annotated by two experts. With 14 participating teams, this challenge can be considered a success. As might be expected, the submitted solutions are based on deep learning. This paper thoroughly evaluates these solutions: in particular, the quality of their annotations are compared to that of human interpretations. Next, lessons learnt from the differential analysis of these solutions are discussed. We expect that they will guide the design of efficient surgery monitoring tools in the near future.

Keywords: cataract surgery, video analysis, deep learning, challenge

1. Introduction

Video recording is a unique solution to collect information about a surgery. Combined with computer vision and machine learning, it allows a wide range of applications, including automatic report generation, surgical skill evaluation and training, surgical workflow optimization, as well as warning and recommendation generation. Key indicators of what the surgeon is doing at

any given time are the surgical tools that he or she is using. Therefore, several tool detection techniques have been presented in recent years [Bouget et al., 2017]. The Challenge on Automatic Tool Annotation for cataRACT Surgery (CATARACTS)[1] was organized in 2017 to evaluate the relevance of these techniques and novel ones in the context of cataract surgery. Cataract surgery is indeed of particular importance: with 19 million surgeries performed annually, it is the most common surgical procedure worldwide [Trikha et al., 2013]. In particular, it is the first surgery that eye surgeons need to master. This paper introduces the results and main conclusions of the CATARACTS challenge.

In recent years, the number of medical image analysis challenges has exploded. According to Grand-Challenge[2], which lists those challenges and hosts some of them, two challenges were organized per year in 2007 and 2008; their number progressively increased to 15 per year in 2012 and 2013; more than 20 challenges are now organized every year. The first challenge organized in the context of ophthalmology was the Retinopathy Online Challenge in 2009 [Niemeijer et al., 2010]: the goal was to detect signs of diabetic retinopathy in fundus photographs. Two other challenges were organized on the same topic: the Diabetic Retinopathy Detection challenge in 2015[3] and the IDRiD challenge in 2018.[4] The detection and segmentation of retinal anomalies in optical coherence tomography images was the topic of three other challenges: the Retinal Cyst Segmentation Challenge in 2015,[5] RETOUCH[6] and ROCC[7] in 2017. However, CATARACTS is the only challenge related to ophthalmic surgery and ophthalmic video analysis. Outside the scope of ophthalmology, three other challenges about surgery video analysis have been organized: EndoVis in 2015 and 2017 [Bernal et al., 2017],[8] and M2CAI in 2016 [Twinanda et al., 2016].[9] Although those three challenges are related to digestive surgery, they share similarities with CATARACTS.

[1] https://cataracts.grand-challenge.org
[2] https://grand-challenge.org/All_Challenges
[3] http://www.kaggle.com/c/diabetic-retinopathy-detection
[4] https://idrid.grand-challenge.org
[5] https://optima.meduniwien.ac.at/research/challenges
[6] https://retouch.grand-challenge.org
[7] https://rocc.grand-challenge.org
[8] https://endovis.grand-challenge.org
[9] http://camma.u-strasbg.fr/m2cai2016

In particular, M2CAI had a sub-challenge on tool detection and both editions of EndoVis had a sub-challenge on tool segmentation. What makes tool detection particularly challenging in CATARACTS, compared to EndoVis and M2CAI, probably is the large range of tools that must be recognized. The reason is that digestive surgeries addressed in EndoVis and M2CAI rely on robotic arms with a standardized set of tools, whereas eye surgeons operate manually and can therefore chose from a wide selection of tools from several manufacturers.

The state-of-the-art solutions for image classification clearly are convolutional neural networks (CNNs) [LeCun et al., 2015]. In the last few years, CNNs have won all image analysis challenges [Russakovsky et al., 2015], including in the medical domain.[10] This success was initiated by AlexNet [Krizhevsky et al., 2012], an 8-layer architecture. AlexNet was quickly superceeded by deeper CNNs, including the 16-layer and 19-layer VGG architectures [Simonyan and Zisserman, 2015]. However, as the depth of CNNs increases, gradients tend to vanish, which makes the backpropagation algorithm inefficient. This problem was efficiently addressed by Inception networks [Szegedy et al., 2016, 2017], residual networks (ResNet) [He et al., 2016a] and dense networks (DenseNet) [Huang et al., 2017]. In Inception networks, auxiliary cost functions are added at the output of several intermediate layers so that gradient backpropagation does not start solely from the end of the network. In ResNet, rather than simply propagating its outputs, each layer propagates the sum of its inputs and of its outputs. Propagating the inputs provides a shortcut for gradient backpropagation. In DenseNet, each layer processes the outputs of all previous layers, as opposed to the outputs of the preceding layer only, which also provides shortcuts for gradient backpropagation. As a result, CNNs with up to 200 and 264 layers can be trained with ResNet and DenseNet, respectively [He et al., 2016b; Huang et al., 2017]. Besides increasing the number of layers, another strategy was investigated in Inception architectures to push performance further: conventional layers are replaced with multi-scale feature extractors, called modules, where convolutions with varying supports are computed in parallel [Szegedy et al., 2016, 2017]. More recently, computer-generated CNN architectures called NASNet were obtained by combining such modules automatically [Zoph et al., 2017].

The task that must be addressed in CATARACTS is more general than

[10] `https://grand-challenge.org`

image classification: class labels must be inferred for each frame, but information from past and/or previous frames in the video can be used to infer those labels. In other words, this is a video labeling task. The 2-D image classification CNNs mentioned above are also very popular in this context [LeCun et al., 2015]. The reason is of course that the main cue for classifying one frame generally is its own visual content. 3-D CNNs have also been proposed for video classification and labeling, in order to take contextual and temporal information into account simultaneously [Ji et al., 2013; Zhu et al., 2016]. However, 2-D CNNs are much more popular. Besides being less computationally intensive, their main advantage is the ability to use transfer learning [Yosinski et al., 2014; Litjens et al., 2017]: image classification models, generally pre-trained on ImageNet[11], can be fine-tuned on individual frames extracted from training videos. This strategy was followed by the winners of M2CAI tool detection sub-challenge [Raju et al., 2016; Sahu et al., 2016; Twinanda et al., 2017; Zia et al., 2016]. Once CNNs are trained, their predictions can be improved using a temporal model. In the simplest scenario, each prediction signal can be smoothed by a usual temporal filter (e.g. a median filter) to compensate for short-term occlusion or image quality problems. Whenever long-term relationships between events are important, a recurrent neural network (RNN) is generally used instead [Yao et al., 2015; Donahue et al., 2017]. CNN+RNN models have thus been used for surgical workflow analysis in endoscopy videos [Twinanda et al., 2017; Jin et al., 2016; Bodenstedt et al., 2017]. Given the correlation between surgical workflow and tool usage, such an approach also seems relevant for tool usage annotation in surgery videos [Mishra et al., 2017; Al Hajj et al., 2017].

The remainder of the paper is organized as follows. The setup of the CATARACTS challenge is described in section 2. Competing solutions are presented in section 3. Results are reported in section 4. The paper ends with a discussion and conclusions in section 5.

2. Challenge Description

2.1. Video Collection

The challenge relies on a dataset of 50 videos of cataract surgeries performed in Brest University Hospital between January 22, 2015 and September 10, 2015. Reasons for surgery included age-related cataract, traumatic

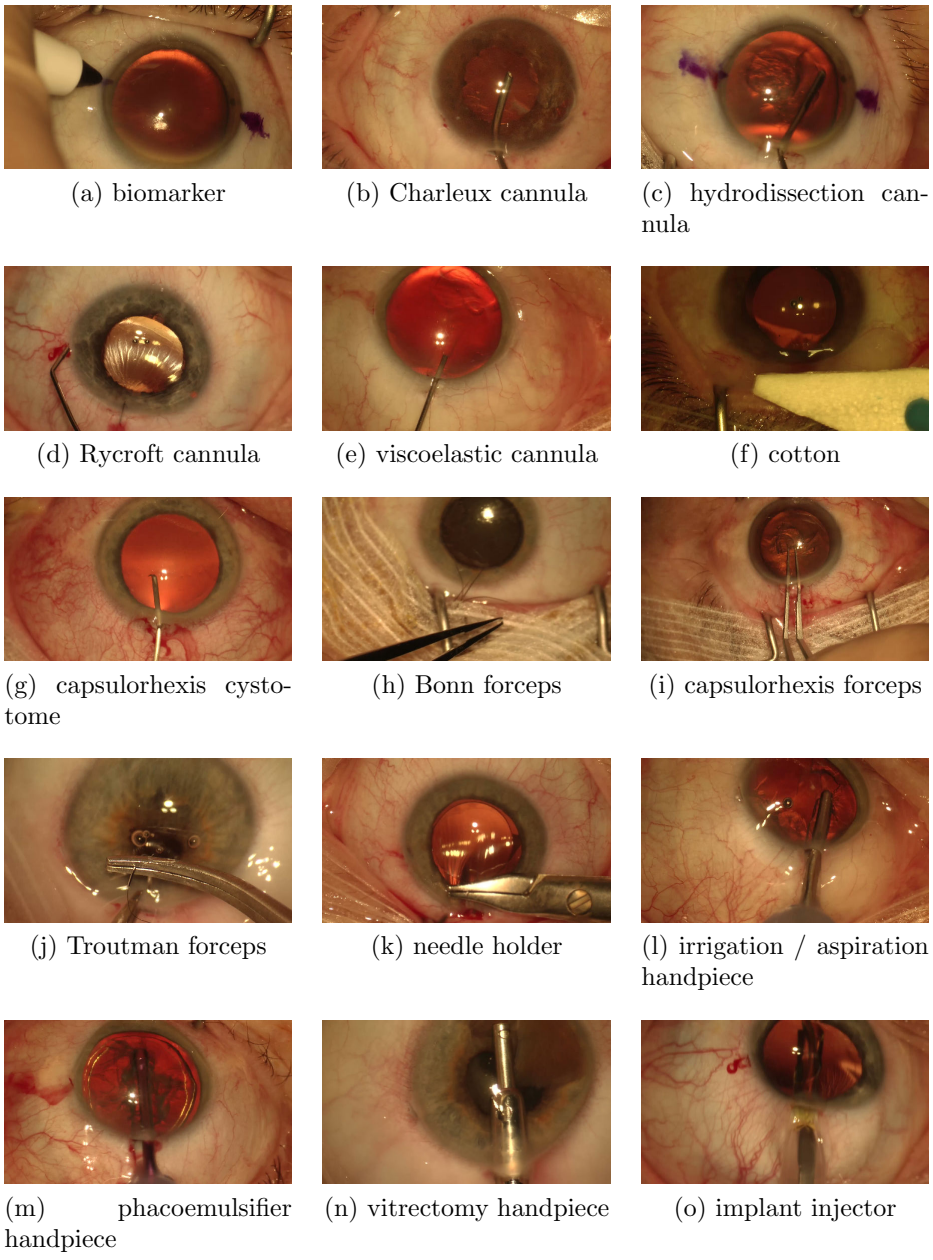
[11] `www.image-net.org`

cataract and refractive errors. Patients were 61 years old on average (minimum: 23, maximum: 83, standard deviation: 10). There were 38 females and 12 males. Informed consent was obtained from all patients. Surgeries were performed by three surgeons: a renowned expert (48 surgeries), a one-year experienced surgeon (1 surgery) and an intern (1 surgery). Surgeries were performed under an OPMI Lumera T microscope (Carl Zeiss Meditec, Jena, Germany). Videos were recorded with a 180I camera (Toshiba, Tokyo, Japan) and a MediCap USB200 recorder (MediCapture, Plymouth Meeting, USA). The frame definition was 1920x1080 pixels and the frame rate was approximately 30 frames per second. Videos had a duration of 10 minutes and 56 s on average (minimum: 6 minutes 23 s, maximum: 40 minutes 34 s, standard deviation: 6 minutes 5 s). In total, more than nine hours of surgery have been video recorded.

2.2. Tool Usage Annotation

All surgical tools visible in microscope videos were first enumerated and labeled by the surgeons: a list of 21 tools was obtained (see Fig 1). Then, the usage of each tool in videos was annotated independently by two non-clinical experts. A tool was considered to be in use whenever it was in contact with the eyeball. Therefore, a timestamp was recorded by both experts whenever one tool came into contact with the eyeball, and also when it stopped touching the eyeball. Up to three tools may be used simultaneously: two by the surgeon (one per hand) and sometimes one by an assistant. Annotations were performed at the frame level, using a web interface connected to an SQL database. Finally, annotations from both experts were adjudicated: whenever expert 1 annotated that tool A was being used, while expert 2 annotated that tool B was being used instead of A, experts watched the video together and jointly determined the actual tool usage. However, the precise timing of tool/eyeball contacts was not adjudicated. Therefore, a probabilistic reference standard was obtained:

- 0: both experts agree that the tool is not being used,
- 1: both experts agree that the tool is being used,
- 0.5: experts disagree.



(a) biomarker (b) Charleux cannula (c) hydrodissection cannula

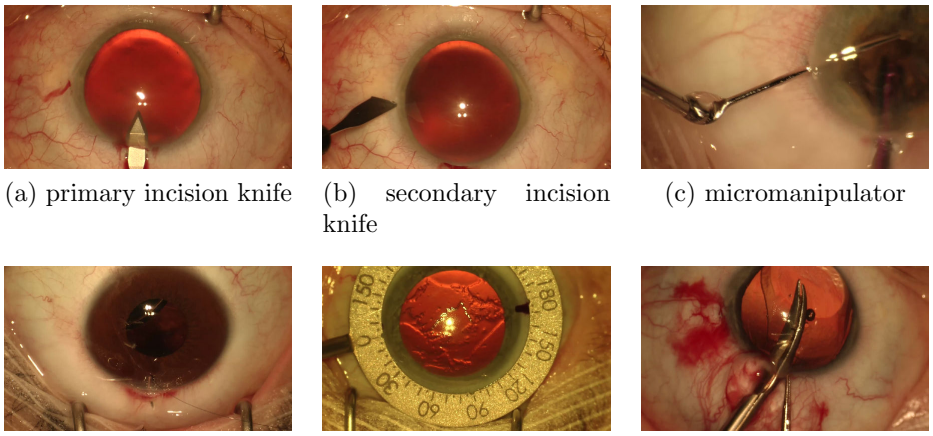
(d) Rycroft cannula (e) viscoelastic cannula (f) cotton

(g) capsulorhexis cystotome (h) Bonn forceps (i) capsulorhexis forceps

(j) Troutman forceps (k) needle holder (l) irrigation / aspiration handpiece

(m) phacoemulsifier handpiece (n) vitrectomy handpiece (o) implant injector

Figure 1: Surgical tools annotated in videos



(a) primary incision knife (b) secondary incision knife (c) micromanipulator

(d) suture needle (e) Mendez ring (f) Vannas scissors

Figure 2: Figure 1 (Cont.).

Inter-rater agreement, before and after adjudication, is reported in Table 1. A chord diagram[12] illustrating the co-occurrence of tools in training video frames is reported in Fig. 3.

2.3. Performance Evaluation

A figure of merit was first computed for each tool label T: the annotation performance for tool T was defined as the area $A_z(T)$ under the receiver-operating characteristic (ROC) curve (see Fig. 6). This curve is obtained by varying a cutoff on the confidence levels for this tool label. Frames associated with a disagreement between experts (reference standard = 0.5 for tool T) were ignored when computing the ROC curve. Then, a global figure of merit was defined: it was simply defined as the mean $A_z(T)$ value over all tool labels T.

The organizers decided to use the area under the ROC curve, rather than figures of merit based on precision and recall, which evaluate cutoffs on the rank of tool labels, sorted by decreasing confidence level. The reason for this choice is that a varying number of tools may be used in each frame (zero, one, two or three). The rank is of limited practical value in this

[12] `http://mkweb.bcgsc.ca/tableviewer/`

Tool	Agreement before adjudication	Agreement after adjudication	% of training frames in use
biomarker	0.835	0.835	0.0168 %
Charleux cannula	0.949	0.963	1.79 %
hydrodissection cannula	**0.868**	**0.982**	2.43 %
Rycroft cannula	**0.882**	**0.919**	3.18 %
viscoelastic cannula	**0.860**	**0.975**	2.54 %
cotton	0.947	0.947	0.751 %
capsulorhexis cystotome	0.994	0.995	4.42 %
Bonn forceps	0.793	0.798	1.10 %
capsulorhexis forceps	0.836	0.849	1.62 %
Troutman forceps	0.764	0.764	0.258 %
needle holder	0.630	0.630	0.0817 %
irrigation/aspiration handpiece	0.995	0.995	14.2%
phacoemulsifier handpiece	0.996	0.997	15.3 %
vitrectomy handpiece	0.998	0.998	2.76 %
implant injector	0.980	0.980	1.41 %
primary incision knife	0.959	0.961	0.700 %
secondary incision knife	0.846	0.852	0.522 %
micromanipulator	0.990	0.995	17.6 %
suture needle	0.893	0.893	0.219 %
Mendez ring	0.941	0.953	0.100 %
Vannas scissors	0.823	0.823	0.0443 %

Table 1: Statistics about tool usage annotation in the CATARACTS dataset. The first two columns indicate inter-rater agreement (Cohen's kappa) before and after adjudication; the largest changes are in bold. The last column indicates the prevalence of each tool in the training subset, ignoring the frames where experts disagree about the usage of that tool, even after adjudication.

scenario: algorithms should not always produce the same number of tool predictions, regardless of the number of tools actually being used. Cutoffs on the confidence level, as used in ROC analysis, are more convenient: a binary prediction can be made independently for each tool label, leading to an adaptive number of tool predictions per frame.

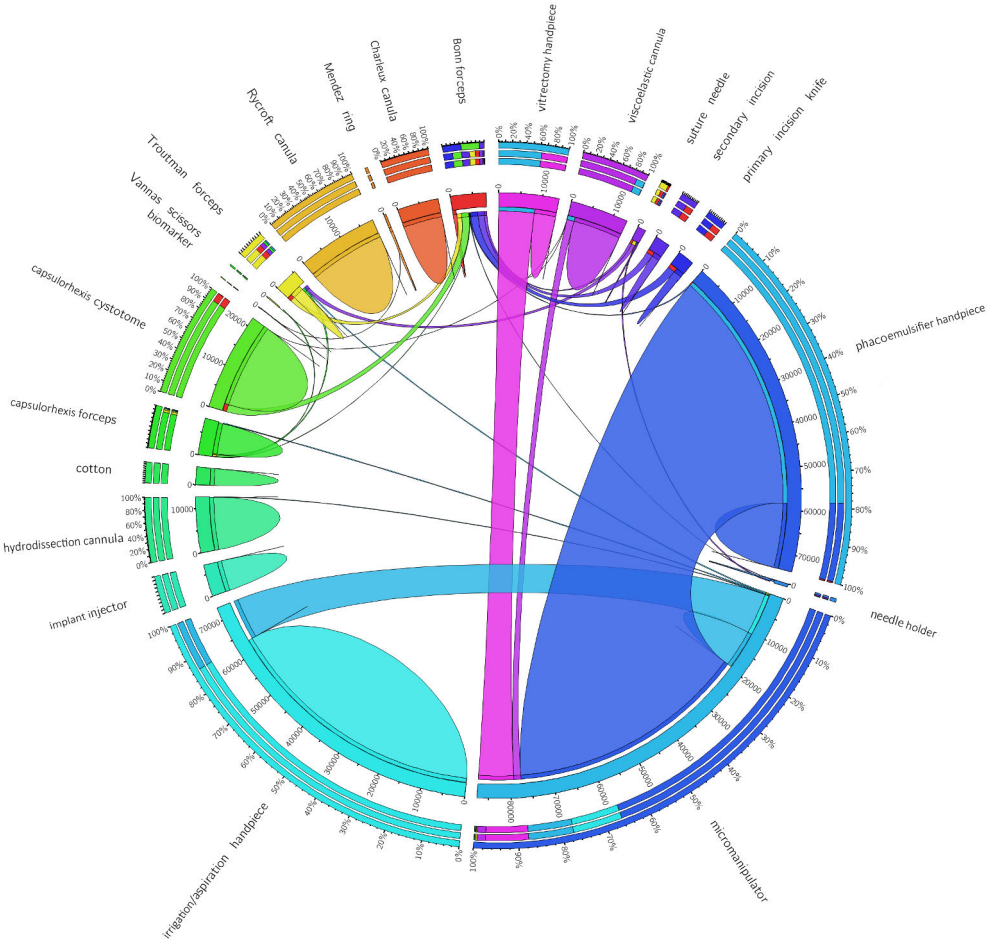


Figure 3: Chord diagram illustrating tool co-occurrence in training video frames. This figure shows, for instance, that the phacoemulsifier handpiece is used in 74,000 frames and that, in 78,5% of these frames, it is used in conjunction with the micromanipulator.

2.4. Rules of the Challenge

The challenge has been continuously accepting submissions during eight months (from April 1, 2017 to November 30, 2017). In order to stimulate competition and to explore more solutions, participants were allowed to submit multiple solutions throughout this period. However, two restrictions were imposed on re-submissions:

1. Each submission was required to be substantially different from the

previous ones. Typically, a first submission may consist of a CNN only, a second one may consist of an ensemble of CNNs, and third one may include a temporal sequencer. However, submitting the same algorithm with different meta-parameters was not allowed. This rule was fixed to minimize the risk of influencing the solution's behavior with test data. To allow verification of this rule by the organizers, a technical report was required for each submission and re-submission.

2. Technical reports and performance scores were immediately published on the challenge website and no re-submission was evaluated for a week. This rule was fixed to balance the inequities between teams submitting multiple solutions and those submitting only once: the latter can benefit from experience gained by the former.

For each team, the solution with maximal performance among all submissions (if more than one) was retained to compile the final team ranking. Two submissions were excluded from the establishment of this ranking by virtue of the one week waiting rule: the scheduled evaluation date occurred after the challenge closing date. However, they are discussed in the following section anyway. Solutions submitted by the organizers (LaTIM) are not included in the team ranking, but are also discussed in this paper.

3. Competing Solutions

Fourteen teams competed in this challenge. Their solutions, as well as the organizers' solution, are described hereafter. To allow comparisons between these solutions, key elements are reported in Tables 2, 3, 4, 5 and 6.

3.1. VGG fine-tuning

The VGG fine-tuning solution uses a CNN with weights pre-trained on the ImageNet dataset. The base network is VGG-16 [Simonyan and Zisserman, 2015]. The last fully connected layer, namely 'fc8', was changed to have twenty-one output neurons, each representing the likelihood that one tool is being used by the surgeon in the input image. The last two fully connected layers, namely 'fc7' and 'fc8', were fine-tuned using the CATARACTS training dataset. The CNN processes images with 288×288 pixels. It was trained using a stochastic gradient descent with a learning rate of 0.001 and a momentum of 0.9. The mini-batch size was set to 48 and the number of epochs to 80. A weighted loss function was used: a weight of one was assigned to

label 0 (tool not being used) and a weight of thirty was assigned to label 1 (tool in use). No random distortions are applied to input images during training and inference.

3.2. LCCV-Cataract

The LCCV-Cataract solution relies on an Inception-v3 CNN [Szegedy et al., 2016] pre-trained on ImageNet. The major difference with other solutions is that a multi-class classifier was trained *(each image has exactly one label)*, rather than a multi-label classifier *(each image may have zero, one or multiple labels)*. Twenty-two mutually exclusive classes were defined: each of the first 21 classes predicts the usage of one tool and the 22$^{\text{nd}}$ class predicts the absence of tool usage. For compatibility reasons, all video frames associated with multiple tools in the CATARACTS dataset were ignored during training. The CNN processes images with 299×299 pixels. It was fine-tuned with a learning rate of 0.01 for several thousand iterations with cross-entropy loss. During inference, the purpose of the 22$^{\text{nd}}$ class it to lower the probability of the other 21 classes when no tool appears to be in use. No random distortions are applied to input images during training and inference.

3.3. AUGSQZNT

The AUGSQZNT solution extends SqueezeNet, a lightweight CNN [Iandola et al., 2016] with weights pre-trained on ImageNet. The proposed architecture starts with three blocks of convolutional layers and then splits into three parts: one part for the 'cannula' set of labels, one part for the 'forceps' set and one part for the rest. The 'forceps' split of the network uses softmax activations while the other two use sigmoid activations. For validation, 5 complete videos and selected frames containing approximately 20% of frames labelled biomarker, needle holder, vitrectomy handpiece and Vannas scissors from 3 videos were kept aside from training. This was to ensure that each label has approximately 15-20% representation in the validation set. The frames were extracted at 10 frames per second although for rare classes, the frames were duplicated up to 50 times after extraction. Afterwards, all frames were augmented using vertical and horizontal flipping and randomly cropping 70%. The CNN was trained using a binary cross entropy loss function with a 80:10:10 weight ratio assigned to each network split. The Adam optimizer [Kingma and Ba, 2015] was used with a learning schedule starting with the learning rate of 0.01 and subsequently dividing by 10 after every 3 epochs with no improvement in validation loss. During inference, 5-fold

test time augmentation is performed by taking the center, top left, top right, bottom right and bottom left patches from each frame in the test dataset. The predictions are averaged across the 5 patches for each frame.

3.4. SurgiToolNet

The SurgiToolNet solution is a deep learning network based on DenseNet-161 [Huang et al., 2017]. The DenseNet-161 model was pre-trained on ImageNet to accelerate the training process. To use the DenseNet-161 network as a multi-label classifier, a Euclidean loss layer was plugged into the end of the network to compute the sum of squares of differences between the predicted output and the ground truth input. The CNN processes images with 224×224 pixels. It was fine-tuned using stochastic gradient descent with a momentum of 0.9. The initial learning rate was set to 0.001, and was divided by 10 after 50,000 iterations. In the deployment process, a binary classification layer was added at the end of this network: this layer is used to threshold the outputs of the fully connected layer and classify them into binary labels $\in \{0, 1\}$, indicating whether or not each tool is being used by the surgeon in the current frame.

3.5. CRACKER

CRACKER uses a frame-wise tool detector, based on a ResNet-34 [He et al., 2016a] pre-trained on ImageNet, followed by field knowledge-based temporal filtering. The optimizer is the SGDR [Loshchilov and Hutter, 2017] and the loss function is the categorical cross entropy log loss.

Frame-wise tool detector: The model was fine-tuned with a 1:2 subsample of the CATARACTS dataset rescaled to 128×128 pixels. First, the top of the network was trained for a fixed number of epochs. Then, the learning rate was reduced by 1/3 at each 1/3 of the network depth. Finally, the entire network was trained until the cross entropy log loss stagnated in the validation set. Test predictions are the result of the average of the model's output over 4 different test-time augmented versions of the frames.

Knowledge-based temporal filtering: First, the temporally sorted predictions are median-filtered with a sliding filter of size 11. For the irrigation/aspiration handpiece, phacoemulsifier handpiece and implant injector, the filter size was set to 101 instead. All signals are then processed based on the surgical procedure: 1) the irrigation/aspiration and vitrectomy handpieces (IA, V, respectively) usually proceed the phacoemulsifier because the latter is used for lens destruction; 2) the implant injector can never come

before IA or V pieces since the implant can only be injected into the eye once the damaged lens has been removed and 3) the Rycroft cannula should not come before IA or V since it is used for refilling the lens in the end of the surgery. With that in mind, the first occurrence of probability$_{IA} > 0.5$ or probability$_{V} > 0.5$ is used for zeroing erroneous predictions of the above-mentioned tools.

3.6. MIL+resnet

The main contribution of the MIL+resnet solution is the decoupling of the initial task into a binary tool detection stage followed by a 21-class classification to determine the tools present on each given frame. The binary tool detection model is based on the Multiple-Instance Learning (MIL) framework [Quellec et al., 2017a]. The MIL assumption was interpreted in this context as follows: image patches are considered as instances, a patch containing (part of) a tool is considered as a positive instance, and a patch with no signs of tool presence is considered as a negative instance. Accordingly, a given image is considered as a bag containing instances. The sole presence of a positive instance is enough to declare the associated bag as containing a tool, whereas in order for a frame/bag to be declared as not containing tools, it must be composed only of negative instances.

In this stage, a standard CNN architecture was employed, namely the Inception-v3 network, with initial weights pre-trained on the ImageNet dataset. In order to deal with patches, the architecture was modified to perform patch-level classification given the full input image. The deeper layers of the Inception-v3 network were discarded, since the receptive field of each layer grows as the network gets deeper. By discarding deeper layers of the network, the receptive field of the output layer can be effectively reduced. The predicted patch labels must then be combined to produce an image-level prediction. In order to follow the standard MIL assumption, patch predictions are merged into a single prediction by means of a max-pooling function.

The binary tool detector was trained on a binarized tool/no-tool version of the provided ground-truth. The resulting model was applied on the test set to retain frames that contained tools. The predictions on test set were temporally smoothed with a trimmed mean filter to add some robustness. Afterwards, a ResNet CNN was trained only on tool-containing frames, in order to learn to classify which were the present tools. This second stage was considered as a standard 21-class multi-label classification problem. Finally, the trained model was applied only to test frames that had been predicted

as containing tools to decide which tools were present at each moment on the videos from the test set.

3.7. ZIB-Res-TS

The framework of the ZIB-Res-TS comprises of three main parts: stratification of the data, a classification model and temporal smoothing as a post-processing step. Since multiple tools can be visible in an image and tool co-occurrence frequency varies within the dataset, label-set sampling [Sahu et al., 2017] was applied to the data to reduce the bias caused by highly frequent tool co-occurrences. This approach relies on stratified sampling based on the co-occurrences of tools as disjoint classes. The model consists of ResNet-50 which was pre-trained on ImageNet and fine-tuned on the CATARACTS dataset by adding a global average pooling and a fully connected layer on top. The task was formulated as a multi-label classification problem with 22 output units, including a no-tool class (i.e. background) as described by [Sahu et al., 2016]. The network was trained using an Adam optimizer with a learning rate of 0.001 for 25 epochs. Assuming that tool usage transitions are smooth, linear temporal smoothing [Sahu et al., 2017] with a window of five frames is applied during inference in order to reduce false positives by suppressing stand alone detections.

3.8. RToolNet

RToolNet is a fine-tuned 50-layer residual network. After pre-training on ImageNet, the first 31 convolutional layers were frozen and only the remainder of the network was fine-tuned on the CATARACTS dataset using a decaying learning rate schedule. Furthermore, the approach makes heavy use of data augmentation to alleviate the strong correlation that is natural between video frames. The network was trained using a stochastic gradient descent with an initial learning rate of 0.05 and a momentum of 0.9. In the second submission, a weighted loss function was introduced which places more emphasis on training examples from under represented classes. This improved results slightly but also made the training more sensitive to inherent randomness, such as the choice of initial weights or training example order. We assume this to be the reason for the strong performance decrease observed for one tool between both submissions and note that this problem could be mitigated using an ensemble of networks trained with different random seeds.

3.9. CDenseNet

CDenseNet is based on DenseNet-169, and the last fully connected layer consists of 21 units for predicting the probability of the corresponding tool usage. To overcome the imbalance of the dataset, besides extracting 6 frames per second, more images were extracted for the rare tools, and a weighted binary softmargin loss function was adopted after converting all '0' labels in ground truth to '-1'. By this way, better performance was obtained for the rare tools, such as biomarker and Vannas scissors. To train the network, a stochastic gradient descent was used with a decreasing learning rate, initialized to 0.05, and a momentum of 0.9. Unlike other solutions, the CNN was not pre-trained on ImageNet: all weights were initialized randomly following a Gaussian distribution. Efficient DenseNet implementation [Pleiss et al., 2017] in PyTorch was used for accelerating the training procedure and improving the parameter utilization.

3.10. TUMCTNet

In the TUMCTNet solution, Inception-v4 was suitably modified and fine-tuned by introducing independent sigmoids as predictors for tool usage and by increasing the input size to 640×360 pixels to maintain the aspect ratio of the surgical video. To handle imbalance within multi-label settings, the co-occurrence of tools was considered for selecting the samples used for training: the label-set stratification proposed by [Sahu et al., 2017] was used, which resulted in 46 label-sets. In addition to balancing the data-set, such an approach also exploits the relationship between tools during the surgery. During the training of the network, data-augmentation including limited random rotation ($\pm 10^{\circ}$), horizontal flipping, random scaling and center-cropping was used. Training relied on on a stochastic gradient descent with a learning rate of 0.001 and a momentum of 0.9. To improve temporal consistency of the results, temporal weighted averaging is performed during inference. An ensemble of two independently trained models is also employed to improve predictions.

3.11. CatResNet

The CatResNet model uses the 152-layer ResNet architecture for multi-label frame classification. The network was initialized with weights pre-trained on the ImageNet dataset and was further fine-tuned using the CATARACTS training videos (22 videos for training and 3 for validation). The videos were sub-sampled at 3 frames per second and half of the frames that do not

feature any tool were discarded to match the frequency of the most common tool class, although the classes were not balanced further. The output of the network is a fully connected layer with 21 nodes with sigmoid activations and it was initialized with a Gaussian distribution with mean 0 and standard deviation 0.01 to be trained from scratch. During training, the input frames were re-shaped to 224×224 pixels and a random horizontal flip and random rotation within 25 degrees with mirror padding was performed to augment the data. The network was trained using stochastic gradient descent with a mini-batch of 8, a learning rate of 0.0001 and a momentum of 0.9 for a total of 10,000 iterations. For the first submission of this model, the predictions rely on the current frame alone and do not incorporate information from any other previous or following frame. A second submission was made which incorporates temporal smoothing as a post-processing step on the CNN predictions using a centered moving average kernel of size 5, however it does not achieve significantly better results.

3.12. TROLIS

The TROLIS solution differs from the competitors in two major aspects: (i) a classical computer vision algorithm is used to detect the biomarker (the rarest tool), and (ii) separate neural networks are trained for the rare tools and the rest. The training set was pruned first: the frames with video artifacts (tearing) were discarded, each 3 frames were averaged, and pixel-wise similar frames were discarded. The tool categories were split into two: six rare tools and the remaining (regular) tools. For the regular tool identification, the average output of two Resnet-50 networks on frames resized to 256×256 pixels and one Resnet-50 network on frames resized to 512×512 pixels was used. These networks were optimized using stochastic gradient descents. For the rare tools, a new dataset was created: it consists of 3,000 (respectively 2,500) frames with (respectively without) rare tool labels. In addition to these frames from the training set, 1,200 frames from the test set, obtained by performing a forward pass using the three Resnet-50 networks, were used as negative samples. One of the networks was fine-tuned on this new dataset, and its output is used for rare tool identification. For the rarest tool (biomarker) detection, a classical computer vision algorithm is applied: it works by finding black blobs (tip of the marker) and white blobs (bulk of the marker) in each frame. It is assumed that the Mendez ring only appears in videos where the biomarker is present. Similarly, it is assumed that the needle holder only appears in videos with suture needle. Moreover, the first

and last 0.5% frames of every test video is clipped. Finally, predictions are time averaged with a window of 45 frames.

3.13. CUMV

The CUMV solution relies on an ensemble of two CNNs with weights pre-trained on ImageNet: ResNet-101 and DenseNet-169. Each network takes as input a single frame from the surgical video, resized to 224×224 pixels, and outputs label predictions for the current frame. Both networks are trained independently with a stochastic gradient descent, using the cross-entropy loss. The learning rate was set to 0.001 for 6,000 iterations and then to 0.0001 for 5,000 iterations. During inference, a gate function [Hu et al., 2017] is used to combine the results of these two networks, which calculates the inner product of the normalized prediction confidences for each kind of tool.

3.14. DResSys

DResSys, developed at D-Wave, uses an ensemble of deep CNN networks to make predictions on individual video frames and then smooths these predictions across frames with a Markov random field. To extract video frames for training of the CNN ensemble, all frames within videos containing the rare tools (e.g. biomarker, Vannas scissors) were used, but in parts of the video with the most common tools, frames were sampled at a rate of only 6 frames/sec. Further, 40,000 frames were randomly selected at uniform rate from amongst training frames that have no tools. This process provided a total of $\sim$100,000 training images.

Frame-level predictors: In the first two submissions a single 50-layer Residual Network was trained and in subsequent submissions Inception-v4 and NASNet-A [Zoph et al., 2017] were trained in addition to ResNet. All parameters were initialized from pre-trained ImageNet models. Images of 540×960 pixels are used for ResNet-50 and Inception-v4, but since NASNet-A is a much larger network requiring much greater GPU memory, 270×480 images are used for this model. The final submission also uses one additional NASNet-A architecture with a larger image size of 337×600 pixels at input. The training data was augmented by randomly horizontally flipping and cropping images. All networks were trained with the Adam optimizer using a sigmoid cross-entropy loss except for the 337×600-pixel NASNet-A model that used a weighted sigmoid cross entropy loss. Training ran for at most 13 epochs with a batch size of 4. The learning rate for each network

was chosen using cross validation. The prediction probability of each trained frame-level CNN is aggregated using a weighted geometric mean in which the weights were set using a grid search over the validation set.

Temporal smoothing: Several smoothing approaches were explored to capture the dependence of tool labels across consecutive frames. The first submissions were based on a simple median filtering method and the last submission includes a Markov random field (MRF) model. The MRF model provides a probability distribution across the time-dependent label space. Assume that $\boldsymbol{y} = \{y_1, y_2, \ldots, y_T\}$ represents the binary label vector for a given tool where $y_t = 1/0$ indicates the presence/absence of the tool in the t^{th} frame. The proposed MRF model has a chain-like structure and defines a conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x}) \sim \exp\big(-E(\boldsymbol{y};\boldsymbol{x})\big)$ for the label vector $\boldsymbol{y}$ given the video $\boldsymbol{x}$ using an energy function $E(\boldsymbol{y};\boldsymbol{x})$ given by

$$E(\boldsymbol{y};\boldsymbol{x}) = \sum_{t=1}^{T} a(s_t) y_t + \frac{w}{2} \sum_{t=1}^{T} \sum_{n \in N(t)} y_t y_n \quad , \tag{1}$$

where $N(t) = \{t-19, t-17, \ldots, t+19\}$ represents the set of neighboring nodes for the t^{th} frame, and provides long-range temporal connectivity. In Eq. (1), $a(s_t)$ is the bias for the t^{th} frame's label which is computed by shifting and scaling the output of the ensemble frame-level prediction score s_t at frame t. The scalar coupling parameter w in Eq. (1) enforces label agreement between neighboring frames. The w parameter and the shift and scale parameters of the linear map $a(s_t)$ were all set by a grid search and are shared for all the 21 tool categories. The MRF model, $p(\boldsymbol{y}|\boldsymbol{x})$, represents the joint probability distribution for all the labels in the temporal domain for a tool. Given this model, the marginal distribution $p(y_t = 1|\boldsymbol{x})$ is computed using a mean-field approximation [Jordan et al., 1999] and the resultant marginal probability is used as the prediction score for the t^{th} frame. Lastly, in order to process videos efficiently, the MRF model is formed in smaller segments of length ∼20,000 frames.

3.15. LaTIM (organizers)

The LaTIM solution relies on an ensemble of CNNs, whose outputs are processed by an ensemble of RNNs. Convolutional and recurrent networks are trained sequentially using a novel boosting technique [Al Hajj et al., 2017]. In a first submission, the CNN ensemble consists of one Inception-v4, one Inception-ResNet-v2 and one o_O network [Quellec et al., 2017b]; the RNN

ensemble consists of one LSTM [Hochreiter and Schmidhuber, 1997] and one GRU [Cho et al., 2014] network. In a second submission, a single CNN is used: NASNet-A. A different ensemble of RNNs, consisting of three LSTMs, is obtained. All networks are trained using the root mean square propagation algorithm. One major difference between both submissions is that RNNs are bidirectional in the first submission and unidirectional in the second, thus allowing online video analysis. Another difference is that a median filter is applied to each prediction signal in the second submission, for short-term temporal smoothing, whereas the RNNs are only used for long-term temporal analysis by design.

4. Results

A total of 27 submissions from 14 teams was received during the challenge period. Additionally, the organizers (LaTIM) submitted two solutions. A timeline of all these submissions is reported in Fig. 4. In order to establish a team ranking, the solution with maximal average AUC from each team was retained. Note that two solutions were evaluated after the challenge period, in virtue of the one week waiting rule: they were not used to establish the team ranking (see section 2.4). The leaderboard is reported in Table 7, together with the average AUCs and the detailed per-tool AUCs published on the CATARACTS website. This table also reports 95% confidence intervals (CIs) on the average AUCs, which were computed as follows: 1) CIs on the per-tool AUCs were computed using DeLong's method [DeLong et al., 1988], 2) their radii were then combined using the root mean square, assuming independence between tools. Each CI was used for a single comparison: is the corresponding solution significantly better than the following solution in the ranking? Results of this test are also reported in Table 7.

Per-tool AUCs are summarized in Fig. 5 using boxplots. Figure 5 (a) summarizes the performance of each solution: it appears that some solutions can detect all tools equally well while others fail for a few tools in particular. Figure 5 (b) summarizes how well each of these tools is detected by competing solutions: it appears that the Charleux cannula, the biomarker, the suture needle, the needle holder and the viscoelastic cannula are particularly challenging. On the contrary, the phacoemulsifier handpiece and the capsulorhexis cystotome are detected well by all solutions. ROC curves for simple and challenging tools are reported in Fig. 6.

team	training data selection	validation set
DResSys	6 frames per second	3 videos
LaTIM	30 frames per second	2 videos
CUMV	6 frames per second	5 videos
TROLIS	*frequent tools (3 CNNs)*: torn frame removal, adaptive frame selection based on pixel differences *rare tools (5 CNNs)*: 4200 negative frames (including 1200 test frames), 2500 positive frames	3 videos
CatResNet	3 frames per second	3 videos
TUMCTNet	0.8 frames per seconds	3 videos
CDenseNet	5 frames per second for frequent tools, 10 frames per second for rare tools	1/3 frames
RToolNet	5 frames per second, after removing 60% of frames without tools	5 videos
ZIB-Res-TS	6 frames per second, with labelset-based sampling [Sahu et al., 2017]	4 videos
MIL+resnet	15 frames per second	1/5 frames
CRACKER	15 frames per second	1/5 frames
SurgiToolNet	15 frames per second	2 videos
AUGSQZNT	10 frames per second	5 videos + selected frames with rare tools in 3 videos
LCCV-Cataract	24 frames per second	1/5 frames
VGG fine-tuning	15 frames per second	5 videos

Table 2: Training data and validation selection in the competing solutions.

team	random hor. flipping	random cropping	random scaling	random rotation	random shifting
DResSys	✓	✓			
LaTIM	✓		✓	✓	✓
CUMV	✓				
TROLIS	✓		✓	✓	✓
CatResNet	✓			✓	
TUMCTNet	✓	✓	✓	✓	
CDenseNet	✓	✓			
RToolNet	✓	✓		✓	
ZIB-Res-TS	✓		✓	✓	✓
MIL+resnet	✓		✓	✓	✓
CRACKER	✓		✓	✓	✓
SurgiToolNet	✓				
AUGSQZNT	✓	✓			
LCCV-Cataract					
VGG fine-tuning					

Table 3: Geometrical data augmentation in the competing solutions

For a deeper understanding of how each of these solutions analyze surgery videos, typical examples of temporal prediction signals are given in Fig. 7. One can easily notice which solutions include temporal smoothing techniques as post-processing steps (see Table 5). Another observation we can make is that the occurrence of false alarms is highly correlated in these signals: this is particularly clear in Fig. 7 (b).

Given the very good classification performance achieved by the top-raking solutions, we wondered whether or not they achieved human-level performance. To answer this question, we evaluated the competing solutions against the annotations of one expert only, before adjudication (see Fig. 8). We observed that the other human grader is always better than all competing solutions, in the sense that his sensitivity/specificity pair is above all ROC curves. A single exception was observed: for cotton usage detection, the DResSys algorithm is slightly better than the first human grader (see Fig. 8 (c)). To evaluate the cost of using automatic annotations rather than

team	SqueezeNet [Iandola et al., 2016]	VGG-16 [Simonyan and Zisserman, 2015]	Inception-v3 [Szegedy et al., 2016]	Inception-v4 [Szegedy et al., 2017]	ResNet-34 [He et al., 2016a]	ResNet-50 [He et al., 2016a]	ResNet-101 [He et al., 2016a]	ResNet-152 [He et al., 2016a]	DenseNet-161 [Huang et al., 2017]	DenseNet-169 [Huang et al., 2017]	NASNet-A [Zoph et al., 2017]	image size	pre-training
DResSys				1		1					2	540×960 (× 2), 270×480, 337×600	ImageNet
LaTIM											1	331×331	ImageNet
CUMV							1			1		224×224	ImageNet
TROLIS						4						256×256 (× 3), 512×512	ImageNet
CatResNet								1				224×224	ImageNet
TUMCTNet				3								640×360 (× 3)	ImageNet
CDenseNet										1		540×960	no
RToolNet						1						540×960	ImageNet
ZIB-Res-TS						1						480×270	ImageNet
MIL+resnet			1		1							256×256 (early training stages: 128×128)	ImageNet
CRACKER					1							128×128	ImageNet
SurgiToolNet									1			224×224	ImageNet
AUGSQZNT	1											360×640	ImageNet
LCCV-Cataract			1									299×299	ImageNet
VGG fine-tuning		1										288×288	ImageNet

Table 4: Convolutional neural networks used in the competing solutions

manual annotations, we computed the relative specificity decrease at equal sensitivity: results are reported in Table 8.

5. Discussion and Conclusions

We have presented the results of CATARACTS, the challenge on automatic tool annotation for cataract surgery. Given the high number of participants (14), we believe this challenge was a success. It is a unique opportunity to learn lessons that will guide the design of efficient surgery monitoring tools in the near future.

First, lessons can be learnt from the challenges noted by participants. All of them pointed out that the distribution of tools is highly unequal (see Fig. 3) and that tools in the same category are often visually similar to one another (cannulae, forceps, etc.). These problems motivated the use of data resampling strategies, to deal with class imbalance, and the design of adequate cost functions. It was also noted that video tearing artifacts appear at regular time intervals in videos. This problem motivated the use of time filtering techniques. Other properties of cataract surgery videos would

team	test data augmentation	temporal smoothing
DResSys		Markov random field
LaTIM		LSTM ($\times$ 3), median filter
CUMV		
TROLIS		average filter
CatResNet		
TUMCTNet	center cropping	weighted average filter
CDenseNet		average filter
RToolNet		
ZIB-Res-TS		linear smoothing [Sahu et al., 2017]
MIL+resnet		rolling trimmed mean
CRACKER	4 versions of frame	median filter, zeroing of impossible predictions
SurgiToolNet		
AUGSQZNT	5 crops of frame	
LCCV-Cataract		
VGG fine-tuning		

Table 5: Post-processing techniques in the competing solutions

probably have been listed as challenges in the pre-deep learning era: uneven illumination, zoom level variations, partial tool occlusion (only the tool tip is visible), and motion and out-of-focus blur. However, none of them were noted by participants: these problems are indeed handled well by CNNs coupled with adequate data augmentation strategies. On the other hand, other specificities of the CATARACTS dataset were exploited by participants to their advantage. First, tool usage generally does not change between consecutive frames. This factor also motivated the use of time filtering techniques. Second, tool usage usually follows precedence rules (e.g. phacoemulsification precedes implant injection) and the rarest tools are generally used in pairs to manage special events: bleeding (the suture needle and the needle holder), asymmetrical implant management (the biomarker and the Mendez ring), etc. These specificities motivated the use of (ad-hoc or general-purpose) temporal sequencers. However, the use of these temporal sequencers was to be used with caution, due to one specific challenge: tools in the same category are sometimes interchangeable. In particular, all forceps may be used to hold the suture needle, not only the 'needle holder'. In fact, one of the team that used recurrent neural networks (TROLIS) noted a performance increase after removing it.

team	resampling	weighted loss	boosting	rare tool detector	co-occurrence analysis
DResSys		✓			
LaTIM			✓		✓
CUMV					
TROLIS				✓	✓
CatResNet					
TUMCTNet		✓			✓
CDenseNet	✓	✓			
RToolNet		✓			
ZIB-Res-TS	✓	✓			✓
MIL+resnet	✓				
CRACKER					✓
SurgiToolNet					
AUGSQZNT	✓				
LCCV-Cataract		✓			
VGG fine-tuning		✓			

Table 6: Strategies for class imbalance in the competing solutions

The above-mentioned properties of the dataset and of the task at hand guided the design of the proposed solutions. Overall, most teams took the following steps to train their solutions: 1) selecting training frames in training videos, 2) downsampling these frames, 3) performing data augmentation, 4) selecting one or several CNNs pre-trained on ImageNet, 5) fine-tuning these CNNs on the selected video frames, through the minimization of a multi-label cost function, 6) optionally training a multi-CNN aggregation function and 7) optionally training a temporal sequencer. Selecting training frames (i.e. ignoring available training samples) and yet performing data augmentation (i.e. generating new training samples) may seem counter-intuitive. However, in many solutions, the decision to discard training frames was motivated by the need to balance classes. As for the general inference strategy, it can be summarized as follows: 1) resizing each test frame, 2) optionally performing data augmentation, 3) processing the resized frame with each CNN, 4) optionally aggregating the CNN predictions and 5) optionally running a tem-

team	DResSys	*LaTIM*	CUMV	TROLIS	CatResNet	TUMCTNet	CDenseNet	RTooIN$_{et}$	ZIB-Res-TS	MIL+resnet	CRACKER	SurgiToolNet	AUGSQZNT	LCCV-Cataract	VGG fine-tuning
rank	1		2	3	4	5	6	7	8	9	10	11	12	13	14
biomarker	0.9988	0.9847	0.9857	0.9026	0.9752	0.8511	0.9825	0.5797	0.9212	0.8018	0.8114	0.8690	0.9701	0.6983	0.5590
Charleux CN	0.9892	0.9836	0.9735	0.9448	0.9490	0.8366	0.8603	0.8771	0.7846	0.8166	0.7814	0.7257	0.6867	0.6724	0.5359
hydrodissection CN	0.9959	0.9873	0.9847	0.9840	0.9811	0.9570	0.9754	0.9717	0.9842	0.9704	0.9679	0.9091	0.9422	0.8743	0.9524
Rycroft CN	0.9980	0.9946	0.9951	0.9924	0.9810	0.9907	0.9891	0.9908	0.9956	0.9791	0.9682	0.9709	0.9432	0.8391	0.6078
viscoelastic CN	0.9865	0.9822	0.9776	0.9822	0.9423	0.9732	0.9349	0.9545	0.9506	0.9253	0.9120	0.9533	0.8248	0.7353	0.8588
cotton	0.9999	0.9986	0.9890	0.9842	0.9816	0.9854	0.9503	0.9759	0.9821	0.9702	0.9869	0.7213	0.9220	0.8893	0.7044
capsulorhexis cystotome	0.9999	0.9998	0.9987	0.9989	0.9976	0.9968	0.9966	0.9976	0.9933	0.9953	0.9911	0.9450	0.9832	0.9151	0.9609
Bonn forceps	0.9972	0.9949	0.9893	0.9942	0.9852	0.9825	0.9454	0.9726	0.9794	0.9574	0.9529	0.8934	0.9300	0.8188	0.8234
capsulorhexis forceps	0.9993	0.9981	0.9890	0.9845	0.9821	0.9879	0.9700	0.9888	0.9869	0.9759	0.9761	0.9779	0.9486	0.8774	0.8399
Troutman forceps	0.9898	0.9974	0.9917	0.9689	0.9752	0.9803	0.9237	0.9656	0.9827	0.9108	0.9020	0.9017	0.8744	0.7963	0.4826
needle holder	0.9945	0.9936	0.9846	0.9839	0.9500	0.9415	0.8859	0.9709	0.9395	0.8893	0.8853	0.9909	0.8990	0.6011	0.4929
irrigation/aspiration HP	0.9988	0.9989	0.9977	0.9976	0.9950	0.9947	0.9926	0.9913	0.9968	0.9925	0.9915	0.9279	0.9745	0.9019	0.9486
phacoemulsifier HP	0.9998	0.9998	0.9990	0.9993	0.9966	0.9969	0.9963	0.9971	0.9994	0.9966	0.9927	0.9526	0.9854	0.9006	0.9712
vitrectomy HP	0.9993	0.9719	0.9943	0.9960	0.9852	0.9924	0.9550	0.9932	0.9726	0.9778	0.9804	0.9958	0.8552	0.9244	0.1725
implant injector	0.9984	0.9939	0.9906	0.9935	0.9828	0.9852	0.9326	0.9644	0.9739	0.9486	0.9590	0.9172	0.9354	0.9108	0.8384
primary IK	0.9999	0.9965	0.9972	0.9933	0.9858	0.9961	0.9779	0.9848	0.9939	0.9801	0.9824	0.9674	0.9471	0.9195	0.8060
secondary IK	0.9997	0.9994	0.9995	0.9984	0.9984	0.9983	0.9911	0.9978	0.9995	0.9936	0.9889	0.9458	0.9632	0.9251	0.8917
micromanipulator	0.9989	0.9978	0.9940	0.9980	0.9897	0.9967	0.9886	0.9912	0.9917	0.9784	0.9710	0.9923	0.9815	0.6452	0.8706
suture needle	0.9987	0.9990	0.9861	0.9915	0.9320	0.9920	0.9420	0.9796	0.9920	0.9295	0.9284	0.7543	0.9383	0.7301	0.5810
Mendez ring	1.0000	0.9980	0.9999	0.9994	0.9966	0.9959	0.9629	0.9814	0.6317	0.9979	0.9986	0.9999	0.7952	0.9886	0.5064
Vannas scissors	0.9972	0.9842	0.9657	0.9182	0.9533	0.9705	0.9625	0.9673	0.9855	0.9893	0.9876	0.9925	0.6841	0.7579	0.4246
score (average AUC)	**0.9971**	**0.9931**	**0.9897**	**0.9812**	**0.9769**	**0.9715**	**0.9579**	**0.9568**	**0.9541**	**0.9513**	**0.9484**	**0.9192**	**0.9040**	**0.8248**	**0.7061**
lower bound of CI	0.9962	0.9923	0.9871	0.9737	0.9739	0.9653	0.9515	0.9481	0.9489	0.9433	0.9419	0.9004	0.8938	0.8123	0.6953
upper bound of CI	0.9981	0.9938	0.9916	0.9887	0.9799	0.9777	0.9643	0.9656	0.9592	0.9592	0.9549	0.9381	0.9142	0.8374	0.7169
better than the next ranked?	yes	yes	yes	no	yes	yes	no	no	no	no	yes	no	yes	yes	n/a

Table 7: Areas under the ROC curve (AUCs) for the retained solution of each team. To compare consecutive solutions in the ranking, 95% confidence intervals (CIs) on the average AUCs are included. HP refers to handpiece, CN refers to cannula and IK refers to incision knife.

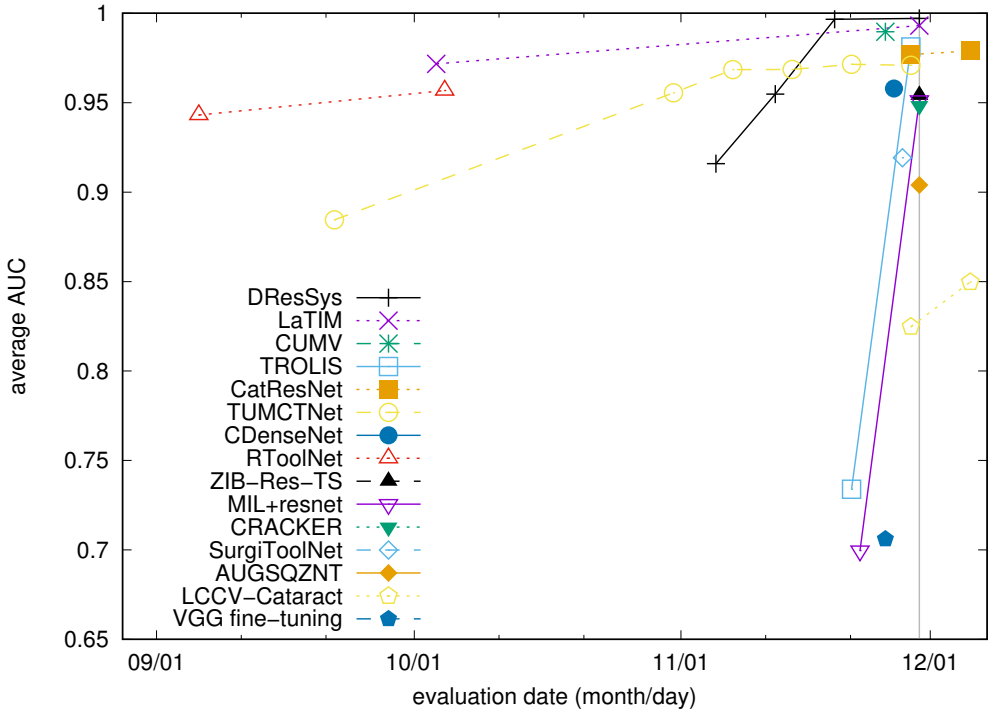
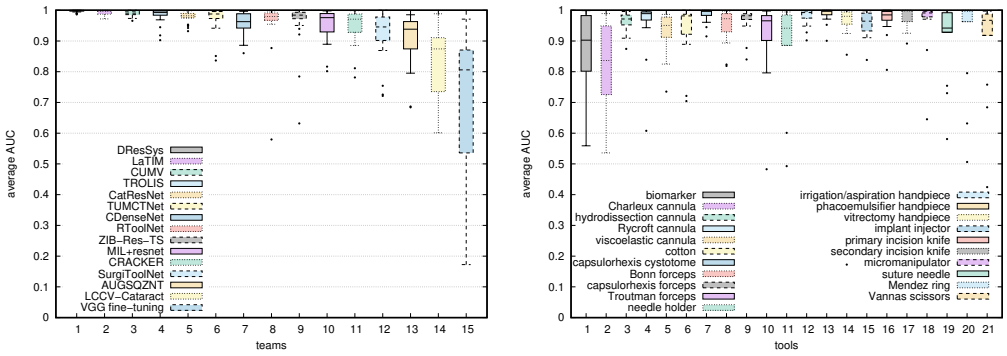


Figure 4: Timeline of solution evaluation — the gray vertical line indicates the challenge closing date. Evaluation dates and submission dates sometimes differed in virtue of the one week waiting rule.

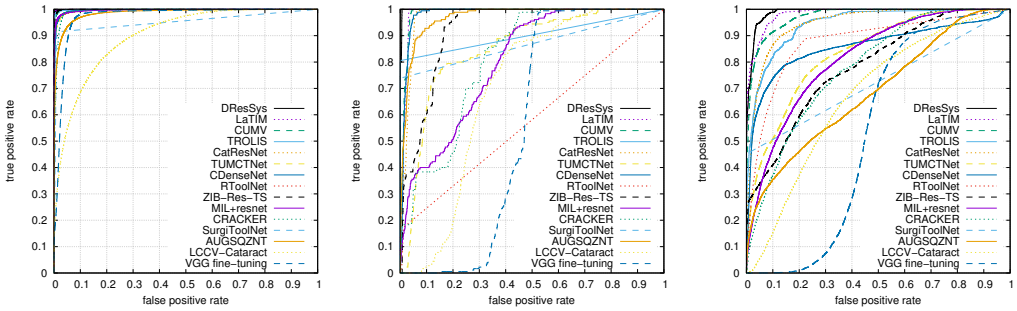
poral filter and/or sequencer. In other words, most participants followed the state-of-the-art approach for multi-label video sequencing using deep learning. It should be noted that no team designed a problem-specific CNN: all solutions relied on CNNs from the literature, with modifications in the final layers only. Beyond these general points, several lessons can be learnt by analyzing the differences between solutions. First, the following factors seem to positively impact the team ranking:

1. keeping full videos aside for validation, as illustrated in Table 2,
2. using data augmentation techniques, as illustrated in Table 3,
3. using the latest generation of CNNs, in particular their deepest versions, as illustrated in Table 4,
4. using multiple CNNs and/or RNNs, as illustrated in Table 4,
5. using temporal smoothing techniques, as illustrated in Table 5 and Fig.



(a) per-team boxplot (b) per-tool boxplot

Figure 5: Boxplots of AUC scores grouped per team or per tool. Each box is drawn around the region between the first and third quartiles, with a horizontal line at the median value. Whiskers extend from the ends of each box to the most distant value which lies within 1.5 times the interquartile range. Black discs indicate outliers.

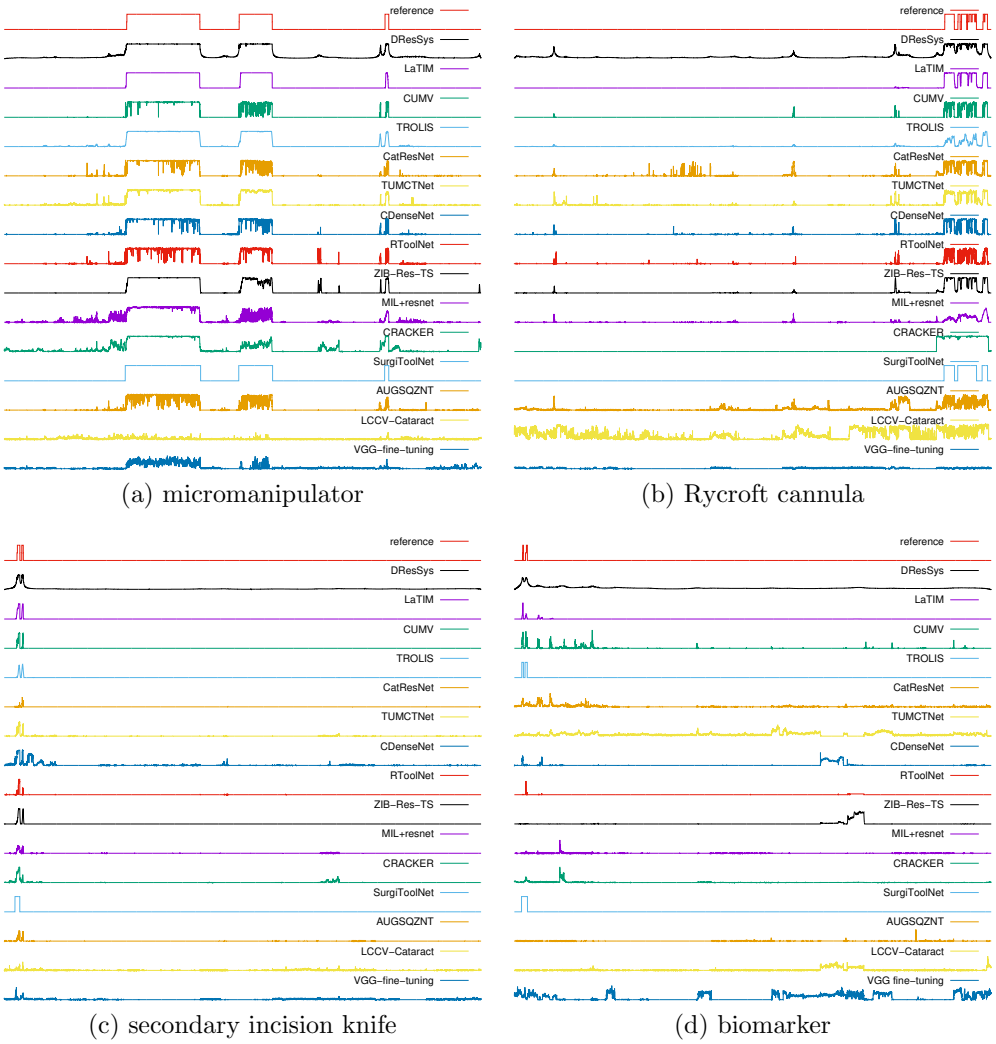


(a) phacoemulsifier handpiece (b) biomarker (c) Charleux cannula

Figure 6: Receiver-operating characteristic (ROC) curves. To save space, ROC curves are reported for three tools only: one frequent and well-detected tool (the phacoemulsifier handpiece) and two challenging tools (the biomarker and the Charleux cannula). Detecting the biomarker is challenging because there are few training samples. Detecting the Charleux cannula is challenging because this tool resembles the Rycroft cannula (in terms of shape and function).

7.

In fact, the winning team (DResSys) combined these five factors. The third

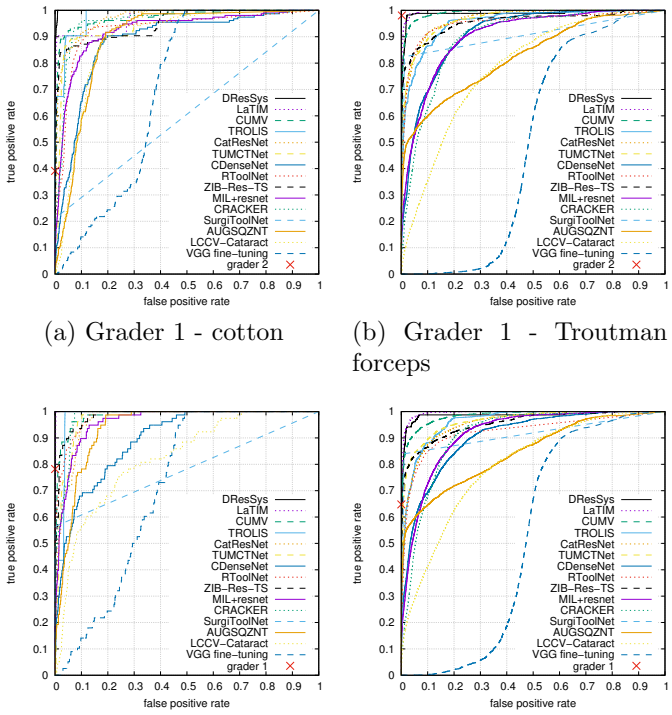


(a) micromanipulator (b) Rycroft cannula

(c) secondary incision knife (d) biomarker

Figure 7: Typical examples of temporal prediction signals. Predictions for the micromanipulator, the Rycroft cannula and the secondary incision knife are from a typical surgery (test video 6). Predictions for the biomarker are from a more complex surgery (test video 13).

lesson seems particularly important: solutions based on the recent NASNet-A architecture achieved top-ranking performance. On the other hand, the following factors do not seem to influence the team ranking: the number



(a) Grader 1 - cotton

(b) Grader 1 - Troutman forceps

(c) Grader 2 - cotton

(d) Grader 2 - Troutman forceps

Figure 8: Receiver-operating characteristic (ROC) curves using the annotations of a single human grader, before adjudication, as reference standard. To save space, ROC curves are reported for two tools only. The sensitivity/specificity pair of the other expert is indicated by a red cross.

of selected training frames (see Table 2), the type of data augmentation (random cropping versus random affine transformations — see Table 3), the CNN's input image size (the CNN's default input size versus a larger size — see Table 4) or the use of test-time data augmentation (see Table 5). If we analyze the specific designs tested by a single team, it can be noted that most of them did not pay. Modeling the tool annotation task as a multi-class classification problem (LCCV-Cataract), rather than a multi-label one, does not work well when more than two tools are used at the same time, which occurs frequently (see Fig. 3). Thresholding predictions as a post-processing step (SurgiToolNet), although important for use in production,

reference	expert 1	expert 2
DResSys	2.93 ± 0.84	1.91 ± 0.72
LaTIM	8.37 ± 2.33	5.58 ± 2.05
CUMV	13.52 ± 2.91	7.53 ± 2.18
TROLIS	19.02 ± 3.84	7.10 ± 2.09
CatResNet	24.74 ± 3.71	13.24 ± 2.81
TUMCTNet	26.15 ± 5.36	16.24 ± 5.32
CDenseNet	41.06 ± 5.55	22.78 ± 5.41
RToolNet	43.61 ± 7.02	26.39 ± 6.66
ZIB-Res-TS	27.97 ± 5.16	18.55 ± 5.08
MIL+resnet	41.36 ± 5.44	24.94 ± 5.25
CRACKER	34.31 ± 4.63	21.88 ± 4.45
SurgiToolNet	67.95 ± 6.95	40.59 ± 9.16
AUGSQZNT	66.13 ± 5.83	42.25 ± 7.61
LCCV-Cataract	68.91 ± 5.38	50.86 ± 5.44
VGG fine-tuning	70.00 ± 3.36	59.51 ± 4.08

Table 8: Relative specificity decrease, compared to the expert, at the same sensitivity. The relative specificity decrease is computed for all 21 tools and the average (± the standard error) is reported.

decreased the solution's merit, evaluated by the area under the ROC curve (see Fig. 6 and 8). The use of a very simple classifier for rare but distinct tools like the biomarker (TROLIS) seemed like a good idea and, it fact, it lead to a very specific classifier (see Fig. 6 (b) and 7 (d)). However, like in the previous example, the use of binary predictions negatively impacted their score. Finally, we note that the most sophisticated solutions (MIL+resnet for instance) did not necessarily rank high, which is disappointing: creativity does not pay well, unless the general training procedure and the five success rules mentioned above are followed (like DResSys).

Compared to most medical image analysis challenges, one of CATARACTS' novelties was to offer participants the ability to submit multiple solutions over a long period of time (8 months). About half of the teams took advantage of this possibility during the last three months of that period (see Fig. 4). Several types of improvements were evaluated: improving data augmentation (tested by TUMCTNet between submissions 1 and 2 — noted "TUMCTNet $1 \rightarrow 2$"), selecting training images differently (DResSys $1 \rightarrow 2$, TROLIS $1 \rightarrow 2$, TUMCTNet $4 \rightarrow 5$ and MIL+resnet $1 \rightarrow 2$), replacing one CNN with another (LaTIM $1 \rightarrow 2$ and TUMCTNet $1 \rightarrow 2$), adding one or several CNNs (DResSys $2 \rightarrow 3$ & $3 \rightarrow 4$, TROLIS $1 \rightarrow 2$ and TUMCTNet $2 \rightarrow 3$

& 3 → 4 & 5 → 6), changing the input size of CNNs (TUMCTNet 4 → 5), redefining training images (DResSys 1 → 2, TROLIS 1 → 2, TUMCTNet 4 → 5 and MIL+resnet 1 → 2), redefining the loss function (DResSys 3 → 4, TUMCTNet 3 → 4, RToolNet 1 → 2, LCCV-Cataract 1 → 2), adding a temporal sequencer (DResSys 1 → 2, CatResNet 1 → 2, TUMCTNet 2 → 3 and MIL+resnet 1 → 2) and replacing this temporal sequencer with another (DResSys 2 → 3). The timeline in Fig. 4 reveals that consecutive submissions almost always led to a performance increase; the only exception was the last submission from the TUMCTNet team, although the decrease was really minor. Increasing performance over time can be explained by the fact that participants progressively increased the complexity of their solution. It also indicates that participants progressively gained experience manipulating the training set and reading other teams' reports. On the down side, allowing multiple submissions introduced one unforeseen training bias: a few teams redefined their validation subset after detailed performance scores in the test set (per-tool AUC) revealed that some of the surgical tools did not appear in their training subset. On one hand, it helped correcting a careless mistake that could have been avoided by frequency counting in the training set. On the other hand, it can be regarded as training on the test set. These submissions were accepted anyway as they also included methodological novelties.

This benchmarking study has one major limitation: solutions were only compared in terms of classification performance, while other aspects are also important. For instance, the ability to analyze tool usage in real-time is of particular interest for the design of intraoperative decision support tools. Some participants (the AUGSQZNT team in particular) decided to design a lightweight solution that would run in real-time with limited hardware, which explains in part a lower ranking compared to those whose did not have that goal in mind. Given the setup of the challenge, it was not possible to compare computation times under identical conditions, so we did not analyze computational aspects in depth. A few lessons can be learnt anyway. First, computation times reported by most participants indicate that their solution can process several frames per seconds using one GPU, which would be enough in many applications. Second, it should be noted that most solutions allow online video analysis, in the sense that they don't need future information for inference. Of course, solutions relying on a symmetrical time filter (see Table 5) would infer predictions with a delay equal to the filter radius. However, this delay is usually less than a second, which would also be acceptable in many applications. Another aspect that would need fur-

ther analysis is the independence on the acquisition hardware: to assess the generality of the proposed solutions, it would be useful to evaluate them on new datasets acquired with different microscopes, different cameras and/or different recorders.

As a final remark, we note that the classification performance of the proposed solutions is lower than that of a human expert (see Fig. 8). However, the performance of top-ranking solutions is very close (see Table 8). Given the limited performance decrease, an automated solution would clearly be a better option, especially in the context of intraoperative decision support: assuming a human interpreter can annotate tool usage in real time, he or she would have to dedicate one hundred percent of his or her time to that task, which would be prohibitive in the long term. Besides, we expect the performance of automated solutions to improve further should contextual information be available. In particular, additional video streams recording the surgical tray or the operating room in general could be considered. In conclusion, the CATARACTS challenge has demonstrated that the task of automated tool annotation in cataract surgery videos has virtually been solved, which paves the way for the introduction of innovative decision support technologies in the operating room, with benefits for both surgeons and patients.

6. Acknowledgments

Teresa Araújo is funded by the FCT grant contract SFRH/BD/122365/2016. Guilherme Aresta is funded by the FCT grant contract SFRH/BD/120435/2016. Adrian Galdran is with ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme, and National Funds through the FCT - Fundação para a Ciência e a Tecnologia within project CMUP-ERI/TIC/0028/2014. Aurélio Campilho is with project "NanoSTIMA: Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics/NORTE-01-0145-FEDER-000016", financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). Manish Sahu, Sabrina Dill, Anirban Mukhopadhyay and Stefan Zachow are funded by German Federal Ministry of Education and Research (BMBF) under the Project BIOPASS (grant Nr. 165V7257).

References

Al Hajj, H., Lamard, M., Conze, P.-H., Cochener, B., and Quellec, G. (2017). Monitoring tool usage in cataract surgery videos using boosted convolutional and recurrent neural networks. *arXiv:1710.01559 [cs]*.

Bernal, J., Tajkbaksh, N., Sánchez, F. J., Matuszewski, B. J., Chen, H., Yu, L., Angermann, Q., Romain, O., Rustad, B., Balasingham, I., Pogorelov, K., Choi, S., Debard, Q., Maier-Hein, L., Speidel, S., Stoyanov, D., Brandao, P., Córdova, H., Sánchez-Montes, C., Gurudu, S. R., Fernández-Esparrach, G., Dray, X., Liang, J., and Histace, A. (2017). Comparative validation of polyp detection methods in video colonoscopy: Results from the MICCAI 2015 endoscopic vision challenge. *IEEE Trans Med Imaging*, 36(6):1231–1249.

Bodenstedt, S., Wagner, M., Katić, D., Mietkowski, P., Mayer, B., Kenngott, H., Müller-Stich, B., Dillmann, R., and Speidel, S. (2017). Unsupervised temporal context learning using convolutional neural networks for laparoscopic workflow analysis. Technical Report arXiv:1702.03684 [cs], Karlsruhe Institute of Technology.

Bouget, D., Allan, M., Stoyanov, D., and Jannin, P. (2017). Vision-based and marker-less surgical tool detection and tracking: a review of the literature. *Med Image Anal*, 35:633–654.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. In *Proc SSST*, pages 103–111, Doha, Qatar. arXiv: 1409.1259.

DeLong, E. R., DeLong, D. M., and Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44(3):837–845.

Donahue, J., Hendricks, L., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., and Darrell, T. (2017). Long-term recurrent convolutional networks for visual recognition and description. *IEEE Trans Pattern Anal Mach Intell*, 39(4):677–691.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proc CVPR*, pages 770–778, Las Vegas, NV, USA.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *Proc ECCV*, Lecture Notes in Computer Science, pages 630–645, Amsterdam, The Netherlands. Springer, Cham.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput*, 9(8):1735–1780.

Hu, X., Yu, L., Chen, H., Qin, J., and Heng, P.-A. (2017). AGNet: Attention-guided network for surgical tool presence detection. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, Lecture Notes in Computer Science, pages 186–194. Springer, Cham.

Huang, G., Liu, Z., Maaten, L. v. d., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proc IEEE CVPR*, pages 2261–2269, Honolulu, HI, USA.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size. Technical Report arXiv:1602.07360 [cs].

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Trans Pattern Mach Intell*, 35(1):221–231.

Jin, Y., Dou, Q., Chen, H., Yu, L., and Heng, P.-A. (2016). EndoRCN: recurrent convolutional networks for recognition of surgical workflow in cholecystectomy procedure video. Technical report, The Chinese University of Hong Kong.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Mach Learn*, 37(2):183–233.

Kingma, D. and Ba, J. (2015). Adam: a method for stochastic optimization. In *Proc ICLR*, San Diego, CA, USA.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Proc Adv Neural Inform Process Syst*, volume 25, pages 1097–1105, Granada, Spain.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Med Image Anal*, 42(Supplement C):60–88.

Loshchilov, I. and Hutter, F. (2017). SGDR: Stochastic gradient descent with warm restarts. In *Proc ICLR*, Toulon, France.

Mishra, K., Sathish, R., and Sheet, D. (2017). Learning latent temporal connectionism of deep residual visual abstractions for identifying surgical tools in laparoscopy procedures. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2233–2240.

Niemeijer, M., Ginneken, B. v., Cree, M. J., Mizutani, A., Quellec, G., Sanchez, C. I., Zhang, B., Hornero, R., Lamard, M., Muramatsu, C., Wu, X., Cazuguel, G., You, J., Mayo, A., Li, Q., Hatanaka, Y., Cochener, B., Roux, C., Karray, F., Garcia, M., Fujita, H., and Abramoff, M. D. (2010). Retinopathy Online Challenge: Automatic detection of microaneurysms in digital color fundus photographs. *IEEE Transactions on Medical Imaging*, 29(1):185–195.

Pleiss, G., Chen, D., Huang, G., Li, T., van der Maaten, L., and Weinberger, K. Q. (2017). Memory-efficient implementation of DenseNets. Technical Report arXiv:1707.06990 [cs].

Quellec, G., Cazuguel, G., Cochener, B., and Lamard, M. (2017a). Multiple-instance learning for medical image and video analysis. *IEEE Rev Biomed Eng*, 10:213–234.

Quellec, G., Charrière, K., Boudi, Y., Cochener, B., and Lamard, M. (2017b). Deep image mining for diabetic retinopathy screening. *Med Image Anal*, 39:178–193.

Raju, A., Wang, S., and Huang, J. (2016). M2CAI surgical tool detection challenge report. Technical report, University of Texas at Arlington.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L.

(2015). ImageNet large scale visual recognition challenge. *Int J Comput Vis*, 115(3):211–252.

Sahu, M., Mukhopadhyay, A., Szengel, A., and Zachow, S. (2016). Tool and phase recognition using contextual CNN features. Technical Report arXiv:1610.08854 [cs.CV], Zuse Institute Berlin.

Sahu, M., Mukhopadhyay, A., Szengel, A., and Zachow, S. (2017). Addressing multi-label imbalance problem of surgical tool detection using CNN. *Int J Comput Assist Radiol Surg*, 12(6):1013–1020.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proc ICLR*, San Diego, CA, USA.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proc AAAI*, pages 4278–4284, San Francisco, CA, USA.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proc IEEE CVPR*, pages 2818–2826, Las Vegas, NV, USA.

Trikha, S., Turnbull, A. M. J., Morris, R. J., Anderson, D. F., and Hossain, P. (2013). The journey to femtosecond laser-assisted cataract surgery: new beginnings or a false dawn? *Eye (Lond)*, 27(4):461–473.

Twinanda, A. P., Mutter, D., Marescaux, J., de Mathelin, M., and Padoy, N. (2016). Single- and Multi-Task Architectures for Tool Presence Detection Challenge at M2cai 2016. Technical Report arXiv:1610.08851 [cs], University of Strasbourg.

Twinanda, A. P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M., and Padoy, N. (2017). EndoNet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Trans Med Imaging*, 36(1):86–97.

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., and Courville, A. (2015). Describing videos by exploiting temporal structure. In *Proc IEEE ICCV*, pages 4507–4515, Santiago, Chile.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *arXiv:1411.1792 [cs]*.

Zhu, W., Hu, J., Sun, G., Cao, X., and Qiao, Y. (2016). A key volume mining deep framework for action recognition. In *Proc IEEE CVPR*, pages 1991–1999, Las Vegas, NV, USA.

Zia, A., Castro, D., and Essa, I. (2016). Fine-tuning deep architectures for surgical tool detection. Technical report, Georgia Institute of Technology.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2017). Learning transferable architectures for scalable image recognition. *arXiv:1707.07012 [cs, stat]*.

Surgical Tool Detection in Cataract Surgery Videos Through Multi-Image Fusion Inside a Convolutional Neural Network

Hassan Al Hajj[a], Mathieu Lamard[b,a], Katia Charrière[a,c], Béatrice Cochener[b,a,d], Gwenolé Quellec[a,*]

[a] *Inserm, UMR 1101, Brest, F-29200 France*
[b] *Univ Bretagne Occidentale, Brest, F-29200 France*
[c] *IMT Atlantique, LaTIM UMR 1101, UBL, Brest, F-29200 France*
[d] *Service d'Ophtalmologie, CHRU Brest, Brest, F-29200 France*

Abstract

The automatic detection of surgical tools in surgery videos is a promising solution for surgical workflow analysis. It paves the way to various applications, including surgical workflow optimization, surgical skill evaluation and real-time warning generation. A solution based on convolutional neural networks (CNNs) is proposed in this paper. Unlike existing solutions, the proposed CNN does not analyze images independently: it analyzes sequences of consecutive images. Features extracted from each image by the CNN are fused inside the network using the optical flow. For improved performance, this multi-image fusion strategy is also applied while training the CNN. The proposed framework was evaluated in a dataset of 30 cataract surgery videos (6 hours of videos). Ten tool categories were defined by surgeons. The proposed system was able to detect each of these categories with a high area under the ROC curve ($0.953 \leq A_z \leq 0.987$). The proposed detector, based on multi-image fusion, was significantly more sensitive and specific than a similar system analyzing images independently ($p = 2.98 \times 10^{-6}$ and $p = 2.07 \times 10^{-3}$, respectively).

*LaTIM - IBRBS - CHRU Morvan - 12, Av. Foch
29609 Brest CEDEX - FRANCE
Tel.: +33 2 98 01 81 29 / Fax: +33 2 98 01 81 24
Email address: gwenole.quellec@inserm.fr (Gwenolé Quellec)

1. Introduction

With the emergence of imaging devices in the operating room, the automated analysis of videos recorded during the surgery is becoming a hot research topic. Potential applications include report generation, surgical workflow optimization, surgical skill evaluation and real-time warning generation [7]. One solution to monitor the surgery is to recognize which tools are being used at each time instant. Therefore, several tool detection techniques have been proposed in recent years [1]. A MICCAI challenge was organized in 2016 for tool presence detection in laparoscopic surgery videos: the best performing methods all relied on deep learning [8, 9, 11, 12]. Those methods relied on transfer learning: CNNs trained for classifying still images, in the ImageNet dataset, were fine-tuned on images extracted from surgery videos. Motion information was not exploited. Deep learning is now often used for video analysis, in particular for human action recognition [5, 2, 10]. Different strategies have been proposed to take advantage of motion, instead of analyzing images independently inside the video stream. One strategy was to regard videos as 3-D images and therefore analyze them with 3-D CNNs [5]. Another strategy was to combine a CNN, analyzing (2-D) images, with a long short-term memory (LSTM) network analyzing the spatial sequencing [2]. A third strategy was to build two CNNs: one to analyze images and another one to analyze the optical flow between consecutive images [10].

In this paper, we propose to use the optical flow in order to exploit spatial redundancies between consecutive images, to finely combine multiple views of the same object. In a previous solution, the optical flow was used to propagate surgical tool segmentations between consecutive frames [4], in order to reduce the number of frames that need to be processed by a CNN. Here, we propose to include the optical flow inside the CNN in order to take advantage of it while training the CNN.

2. Cataract Surgery Dataset

This study focuses on cataract surgery, the most common eye surgery. A dataset of 30 cataract surgery videos, recorded in 2011 at Brest University Hospital, was used. Surgeries were performed by four different surgeons and lasted 12 minutes on average. Videos were recorded, at the output of the microscope, in interlaced DV format. The frame definition is 576×720 pixels and the frame rate is 25 fps. Each video was then manually annotated

Table 1: Tool Usage

tool category	total duration (seconds)
knife	386
viscous fluid injector	515
clamp	2105
needle	652
cannula	1698
micromanipulator	4477
phacoemulsifier handpiece	4357
irrigation/aspiration handpiece	3901
implant injector	349
cotton	423

by one surgeon, who indicated the appearance and disappearance of each surgical tool in or from the field of view. Surgeons do not always use the same tools from one surgery to another: for instance, the size of tools may change. Therefore, tools were grouped into ten categories, which appear in all videos. Statistics on tool usage are given in table 1. Note that two tools may be visible at the same time in the field of view (one per hand).

3. Proposed Solution

Unlike existing solutions, the proposed detector does not process images independently: it processes sequences of N consecutive images simultaneously. One CNN was trained to detect all tools visible in each sequence as summarized in the section 3.1 and detailed in sections 3.2 to 3.7.

3.1. Transfer Learning from an Image Model to an Image Sequence Model

Processing image sequences is more computationally expansive than processing single images, so training such a CNN also is a priori. To speed training up, a particular kind of transfer learning is proposed in this paper (see Fig. 1):

1. First, as usually done in the field, one CNN is trained using independent images ($N = 1$): this CNN is trained to detect all tools visible in each image.

2. This CNN is then modified as described in section 3.6 in order to process image sequences. The modified network simply adds an additional operator, so all network parameters (weights and biases) are compatible with the above CNN: those parameters are fine-tuned using image sequences.

3.2. Preprocessing and Data Augmentation

To reduce computation times and remove interlacing artifacts, which largely distort the shape of surgical tools, images are first downsampled by a factor of two (definition: 288×360 pixels). For data augmentation purposes, random rotation, translation and scaling operations are applied to each image at each training epoch. In the case of image sequences, the same random rotation, translation and scaling operations are applied to all images in the sequence (see Fig. 2).

3.3. Optical Flow Computation

To allow multi-image fusion inside the CNN, the optical flow is computed from each image I_n in a sequence I to the last image I_N (see Fig. 2). A dense optical flow based on polynomial expansions is used [3]. Let $\Delta^{hor}_{n,x,y}$ and $\Delta^{ver}_{n,x,y}$ denote the horizontal and vertical components of this flow, evaluated at each (x, y) spatial coordinate. Because of data augmentation, the optical flow is recomputed at each training epoch.

3.4. Convolutional Neural Network

The optimal network structure has not been investigated in this preliminary study: we simply reused a CNN structure detailed in our recent publication [6]. The network structure is rather standard (see Table 2). The first layers consist of a succession of convolutional layers followed by a max pooling layer or a root-mean square pooling layer. Those layers are followed by two dense layers, each of which are preceded by a dropout layer and followed by a maxout layer to avoid overfitting. Finally, one dense layer with ten neurons is used to detect the ten tool categories (one neuron per tool category).

3.5. Fusion Inside the Convolutional Neural Network

This CNN is used for processing image sequences as illustrated in the bottom part of Fig. 1:

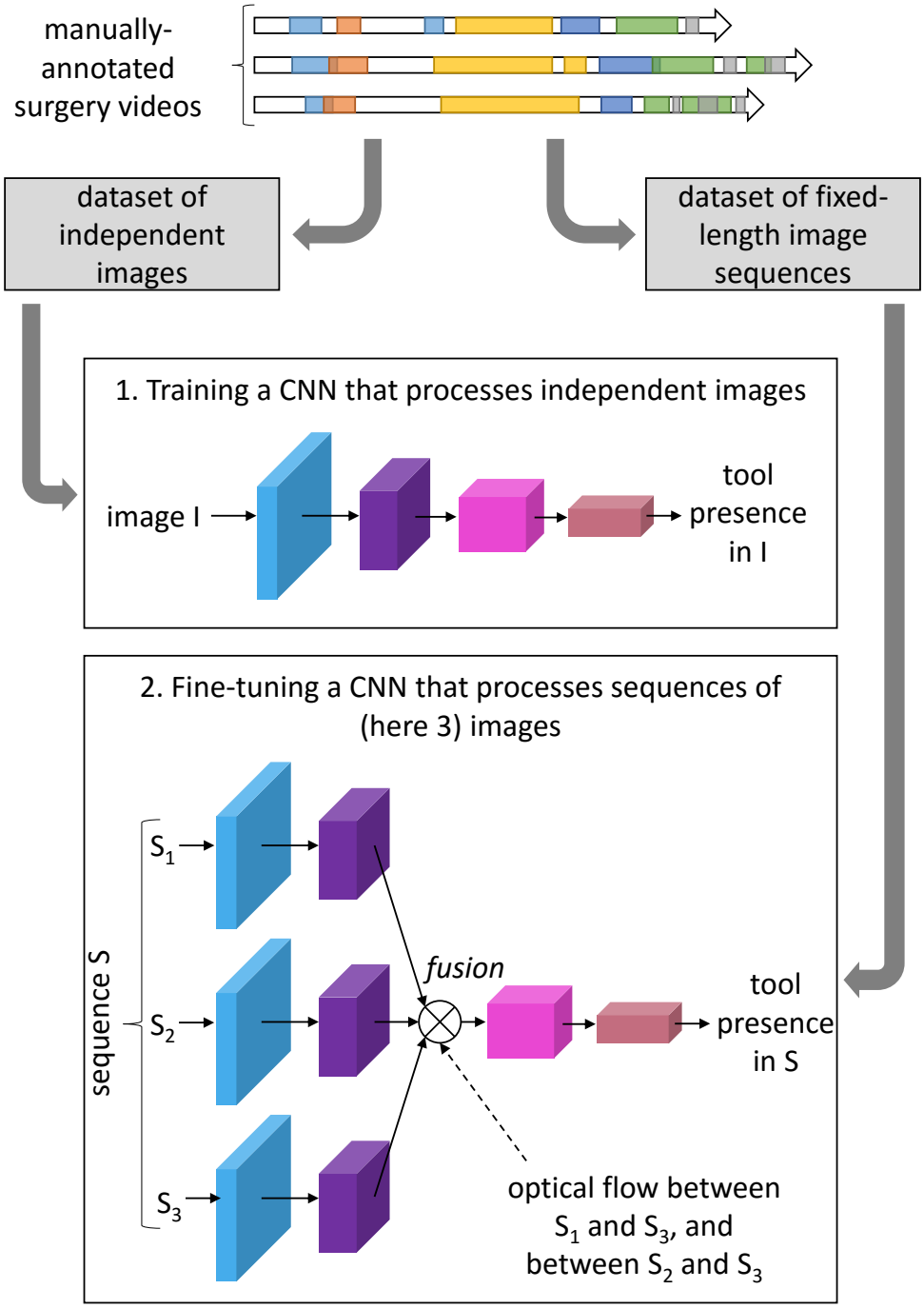


Figure 1: Overview of the framework

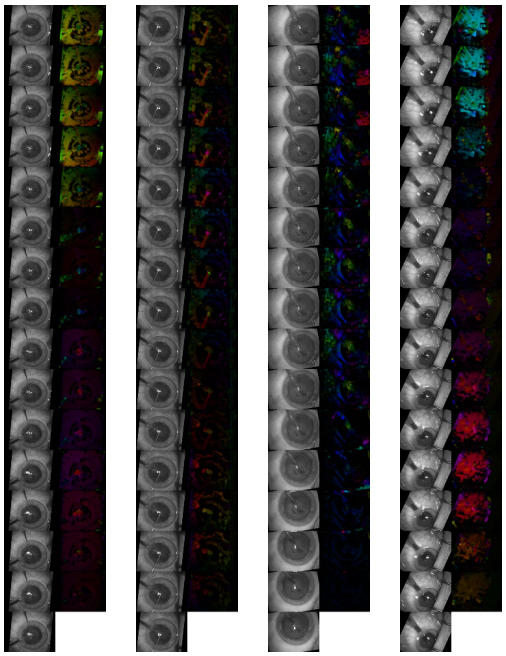


Figure 2: Examples of image sequences, after data augmentation preprocessing. For each sequence, the second column shows the optical flow between the image on the left and the last image of the sequence. The intensity is proportional to motion amplitude and the hue indicates motion direction.

Table 2: Network Structure

id	layer type	activation maps	window size	window stride	output tensor size
1	Input				288 x 360
2	Conv	32	4 x 4	2	144 x 180
3	Conv	32	4 x 4	1	145 x 181
4	MaxPool	32	3 x 3	2	72 x 90
5	Conv	64	4 x 4	2	36 x 45
6	Conv	64	4 x 4	1	37 x 46
7	Conv	64	4 x 4	1	36 x 45
8	MaxPool	64	3 x 3	2	17 x 22
9	Conv	128	4 x 4	1	18 x 23
10	Conv	128	4 x 4	1	17 x 22
11	Conv	128	4 x 4	1	18 x 23
12	MaxPool	128	3 x 3	2	8 x 11
13	Conv	256	4 x 4	1	9 x 12
14	Conv	256	4 x 4	1	8 x 11
15	Conv	256	4 x 4	1	9 x 12
16	MaxPool	256	3 x 3	2	4 x 5
17	Conv	512	4 x 4	1	3 x 4
18	RMSPool	512	3 x 3	2	2 x 2
19	Dropout	$\emptyset$			
20	Dense	1024			
21	Maxout	512			
22	Dropout	$\emptyset$			
23	Dense	1024			
24	Maxout	512			
25	Dense	10			

1. Each image is processed independently up to a given layer: a set of M activation maps is obtained per image, where M is the number of channels in that layer.
2. Those activation maps are then fused per channel: a set of M fused activation maps is obtained.
3. Finally, those fused activation maps are processed with the remaining layers of the CNN.

The complexity of the fusion operator depends on where it is inserted in the network. If it is inserted after a dense layer (*late fusion*), then activation maps contain a single pixel. So a simple fusion operator, like the average or the maximum of the N activation values, can be used. Initial experiments have suggested that the average is a better fusion operator than the maximum ($A_z = 0.969 \pm 0.002$, as opposed to $A_z = 0.964 \pm 0.002$). If, on the other hand, the fusion operator is inserted after a convolutional or a pooling layer (*early fusion*), then the optical flow needs to be taken into account to correctly match activation values between the N maps. Let I denote an image sequence indexed by an image index n, spatial indices x and y, and a channel index c. The following fusion operator was defined for early fusion:

$$\begin{cases} avg(I)_{x,y,c} = \frac{1}{count(I)_{x,y}} \sum\limits_{\substack{n=1..N,\\ u+\Delta^{hor}_{n,u,v}=x,\\ v+\Delta^{ver}_{n,u,v}=y}} I_{n,u,v,c} \\ count(I)_{x,y} = \sum\limits_{\substack{n=1..N,\\ u+\Delta^{hor}_{n,u,v}=x,\\ v+\Delta^{ver}_{n,u,v}=y}} 1 \end{cases} \quad (1)$$

3.6. Gradient of the Fusion Operator

In order to fine-tune the network, a gradient function *avg_grad* must be defined for the above operator. Let f denote the function optimized by the network. To allow backpropagation of the errors, gradient functions must express $\frac{\partial f}{\partial I}$, the partial derivative of f with respect to the operator's input I, as a function of $\frac{\partial f}{\partial J}$, the partial derivative of f with respect to the operator's output J, and possibly of the operator's input I and output J:

$$\begin{cases} avg\_grad\left(\frac{\partial f}{\partial J}, I\right)_{n,x,y,c} = count(I)_{u,v} \frac{\partial f}{\partial J}_{n,u,v,c} \\ u = x + \Delta^{hor}_{n,x,y} \\ v = y + \Delta^{ver}_{n,x,y} \end{cases} \quad (2)$$

Table 3: Influence of the fusion Level. Experiments were performed on one cross-validation fold only, using sequences of $N = 16$ images. Layer identifiers refer to Table 2.

fusion between layers...	*early fusion*					*late fusion*		
	4 & 5	8 & 9	12 & 13	16 & 17	18 & 19	21 & 22	24 & 25	or after layer 25
average A_z	0.964	0.971	0.975	0.977	0.976	0.975	0.975	0.975

3.7. Implementation Details

In order to reuse deep learning code with minimal modifications, sequences of N consecutive images were stored as large images of $288N \times 360$ pixels. Two C++ TensorFlow modules based on OpenCV were defined: one for data augmentation and one for optical flow computation, as well as the *avg* and *avg_grad* operators.

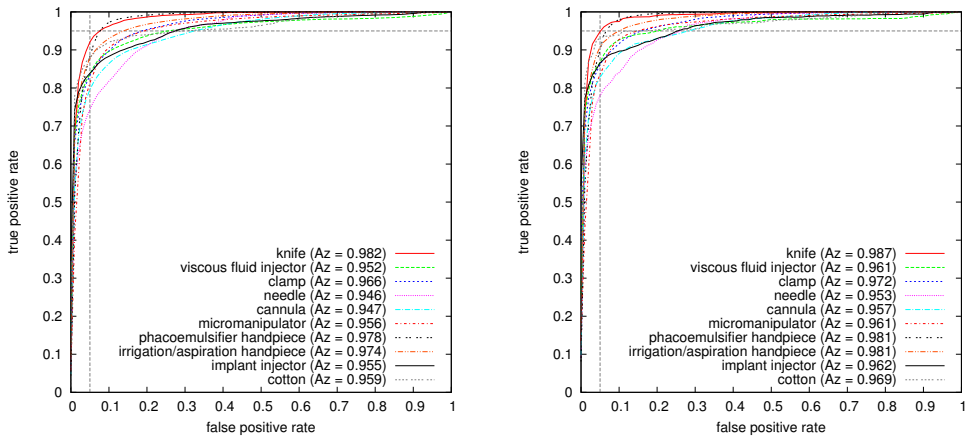
4. Experiments

Sequences of $N = 16$ consecutive images were considered in this study (duration: 0.64 seconds). A total of 531,743 independent images and 531,293 images sequences were extracted from the dataset. The proposed system was validated by cross-validation. In that purpose, the dataset was divided into five groups of six videos and five CNNs were trained: each CNN was trained on four groups and tested on the remaining group.

The tool detection performance in independent images is reported in Fig. 3 (a). Performance of information fusion as a function of the fusion level is reported in table 3: detection performance is optimal at intermediate levels, between layers 16 and 17 in particular (see Table 2). A detailed analysis of detection performance at the optimal fusion level is reported in Fig. 3 (b). On average, to achieve a specificity of 95%, sensitivities of 84.2% and 87.3% are obtained using independent images and image sequences, respectively ($p = 2.98 \times 10^{-6}$). To achieve a sensitivity of 95%, average specificities of 77.2% and 82.0% are obtained using independent images and image sequences, respectively ($p = 2.07 \times 10^{-3}$).

5. Discussion and Conclusions

To our knowledge, this was the first attempt to detect surgical tools in cataract surgery videos using deep learning. The main result is that,



(a) Independent images (b) Sequences of 16 consecutive images

Figure 3: Detection performance for independent images and for image sequences (fusion between layers 16 & 17 — see Table 2). Each ROC curve reported on these figures is actually an average curve, computed over the five cross-validation folds.

although the analysis of independent images is already very efficient, multi-image fusion significantly improves detection performance further. Besides, with a proper data management, the cost of multi-image fusion is limited at test time: while processing a video stream, only the current image needs to be processed by the CNN up to the fusion layer. So, besides optical flow computing, processing image sequences is not more expansive than processing individual images. In future work, we will investigate the influence of N (the length of each image sequence) and of the network structure on tool detection performance. But the performance is already high enough to serve as reliable basis for computer-aided intervention.

References

[1] Bouget, D., Allan, M., Stoyanov, D., and Jannin, P. (2017). Vision-based and marker-less surgical tool detection and tracking: a review of the literature. *Med Image Anal*, 35:633–654.

[2] Donahue, J., Hendricks, L., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., and Darrell, T. (2017). Long-term recurrent convolutional

networks for visual recognition and description. *IEEE Trans Pattern Anal Mach Intell*, 39(4):677–691.

[3] Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Proc. SCIA*, pages 363–370, Halmstad, Sweden.

[4] García-Peraza-Herrera, L. C., Li, W., Gruijthuijsen, C., Devreker, A., Attilakos, G., Deprest, J., Vander Poorten, E., Stoyanov, D., Vercauteren, T., and Ourselin, S. (2016). Real-Time segmentation of non-rigid surgical tools based on deep learning and tracking. Technical report, University College London.

[5] Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Trans Pattern Mach Intell*, 35(1):221–231.

[6] Quellec, G., Charrière, K., Boudi, Y., Cochener, B., and Lamard, M. (2017). Deep image mining for diabetic retinopathy screening. *Med Image Anal.* in press.

[7] Quellec, G., Lamard, M., Cochener, B., and Cazuguel, G. (2015). Real-time task recognition in cataract surgery videos using adaptive spatiotemporal polynomials. *IEEE Trans Med Imaging*, 34(4):877–887.

[8] Raju, A., Wang, S., and Huang, J. (2016). M2CAI surgical tool detection challenge report. Technical report, University of Texas at Arlington.

[9] Sahu, M., Mukhopadhyay, A., Szengel, A., and Zachow, S. (2016). Tool and phase recognition using contextual CNN features. Technical Report arXiv:1610.08854 [cs.CV], Zuse Institute Berlin.

[10] Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Proc NIPS*, volume 27, Montreal, Canada.

[11] Twinanda, A. P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M., and Padoy, N. (2017). EndoNet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Trans Med Imaging*, 36(1):86–97.

[12] Zia, A., Castro, D., and Essa, I. (2016). Fine-tuning deep architectures for surgical tool detection. Technical report, Georgia Institute of Technology.

Université de Bretagne Occidentale

Titre : Analyse vidéo pour la chirurgie de la cataracte augmentée

Mots clés : Chirurgie de la cataracte, détection des outils chirurgicaux, analyse vidéo, réseaux de neurones récurrents et à convolutions

Résumé : L'ère numérique change de plus en plus le monde en raison de la quantité de données récoltées chaque jour. Le domaine médical est fortement affecté par cette explosion, car l'exploitation de ces données est un véritable atout pour l'aide à la pratique médicale. Dans cette thèse, nous proposons d'utiliser les vidéos chirurgicales dans le but de créer un système de chirurgie assistée par ordinateur. Nous nous intéressons principalement à reconnaître les gestes chirurgicaux à chaque instant afin de fournir aux chirurgiens des recommandations et des informations pertinentes. Pour ce faire, l'objectif principal de cette thèse est de reconnaître les outils chirurgicaux dans les vidéos de chirurgie de la cataracte. Dans le flux vidéo du microscope, ces outils sont partiellement visibles et certains se ressemblent beaucoup. Pour relever ces défis, nous proposons d'ajouter une caméra supplémentaire filmant la table opératoire. Notre objectif est donc de détecter la présence des outils dans les deux types de flux vidéo : les vidéos du microscope et les vidéos de la table opératoire. Le premier enregistre l'œil du patient et le second enregistre les activités de la table opératoire. Deux tâches sont proposées pour détecter les outils dans les vidéos de la table : la détection des changements et la détection de présence d'outil. Dans un premier temps, nous proposons un système similaire pour ces deux tâches. Il est basé sur l'extraction des caractéristiques visuelles avec des méthodes de classification classique. Il fournit des résultats satisfaisants pour la détection de changement, cependant, il fonctionne insuffisamment bien pour la tâche de détection de présence des outils sur la table. Dans un second temps, afin de résoudre le problème du choix des caractéristiques, nous utilisons des architectures d'apprentissage profond pour la détection d'outils chirurgicaux sur les deux types de vidéo. Pour surmonter les défis rencontrés dans les vidéos de la table, nous proposons de générer des vidéos artificielles imitant la scène de la table opératoire et d'utiliser un réseau de neurones à convolutions (CNN) à base de patch. Enfin, nous exploitons l'information temporelle en utilisant un réseau de neurones récurrent analysant les résultats de CNNs. Contrairement à notre hypothèse, les expérimentations montrent des résultats insuffisants pour la détection de présence des outils sur la table, mais de très bons résultats dans les vidéos du microscope. Nous obtenons des résultats encore meilleurs dans les vidéos du microscope après avoir fusionné l'information issue de la détection des changements sur la table et la présence des outils dans l'œil.

Title : Video analysis for augmented cataract surgery

Keywords : Cataract surgery, surgical tool detection, video analysis, convolutional and recurrent neural networks

Abstract: The digital era is increasingly changing the world due to the sheer volume of data produced every day. The medical domain is highly affected by this revolution, because analysing this data can be a source of education/support for the clinicians. In this thesis, we propose to reuse the surgery videos recorded in the operating rooms for computer-assisted surgery system. We are chiefly interested in recognizing the surgical gesture being performed at each instant in order to provide relevant information. To achieve this goal, this thesis addresses the surgical tool recognition problem, with applications in cataract surgery. The main objective of this thesis is to address the surgical tool recognition problem in cataract surgery videos. In the surgical field, those tools are partially visible in videos and highly similar to one another. To address the visual challenges in the cataract surgical field, we propose to add an additional camera filming the surgical tray. Our goal is to detect the tool presence in the two complementary types of videos: tool-tissue interaction and surgical tray videos. The former records the patient's eye and the latter records the surgical tray activities.
.

Two tasks are proposed to perform the task on the surgical tray videos: tools change detection and tool presence detection. First, we establish a similar pipeline for both tasks. It is based on standard classification methods on top of visual learning features. It yields satisfactory results for the tools change task, howev-lateer, it badly performs the surgical tool presence task on the tray. Second, we design deep learning architectures for the surgical tool detection on both video types in order to address the difficulties in manually designing the visual features. To alleviate the inherent challenges on the surgical tray videos, we propose to generate simulated surgical tray scenes along with a patch-based convolutional neural network (CNN). Ultimately, we study the temporal information using RNN processing the CNN results. Contrary to our primary hypothesis, the experimental results shows deficient results for surgical tool presence on the tray but very good results on the tool-tissue interaction videos. We achieve even better results in the surgical field after fusing the tool change information coming from the tray and tool presence signals on the tool-tissue interaction videos.