Generation of Artificial Image and Video Data for Medical Deep Learning Applications

Zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

von der KIT-Fakultät für

Elektrotechnik und Informationstechnik

des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Andreas Wachter, Dipl.-Inf.

geb. in Worms

Tag der mündlichen Prüfung:	29. November 2021
Referent:	Prof. Dr. rer. nat. Werner Nahm
Korreferent:	PD Dr. Ing. habil. Thomas Wittenberg

Abstract

In recent years, neural networks (NNs) have achieved remarkable results in event recognition in medical image and video analysis. One of the main limitations of machine learning approaches is the lack of available annotated training data. This lack refers to the number of available datasets and the number of image and video variations in existing datasets. Especially in the medical field, it is hard to extend the number of datasets. The reasons for this are various. For example, legal issues may prevent the publication of the data, or the occurrence of a disease is very rare, making it hard to record it. Moreover, experts must annotate medical data in a time-consuming process and therefore it is expensive. Existing image data augmentation methods are often applied to the video domain. However, these methods are not created sufficiently independent video samples from a dataset. Therefore, it is necessary to develop methods to extend the video dataset retrospectively or generate new synthetic data.

In this thesis, two novel methods are introduced, explained and evaluated by applying them to three medical applications in the field of surgery. First, the workflow augmentation method is introduced, which uses semantic information, e.g., events of a surgical workflow, to augment video data in the temporal domain. The workflow augmentation allows the creation of independent videos from an existing dataset. The proposed method is highly flexible and allows, furthermore, to balance the surgical phases or surgical instruments in a dataset. By applying the method on exemplary medical datasets in cataract surgery, and in laparoscopic cholecystectomy surgery, the method's performance is verified. For instrument recognition, in the example of the cataract surgery, an increase in accuracy (ACC) of 2.8 % to 93.5 % could be achieved compared to established methods. For phase recognition, in the cholecystectomy surgery example, an increase in ACC of 8.7 % to 96.96 % could be achieved compared to a previous study from literature. Both studies impressively demonstrate the potential of the workflow augmentation method.

The second method is based on a generative adversarial network (GAN) approach. It allows the creation of synthetic images. This approach is auspicious when only a few data are available, and new data must be created. In the context of this thesis, cycle generative adversarial networks (CycleGANs) are used to perform an image-to-image translation. Additionally, it is possible to apply conditions to the transformation. The CycleGAN was used in the third study to estimate a facial image of the patient after cranio-maxillofacial surgery using a preoperative portrait photo and the 3D surgical planning model. Thereby,

it was possible to estimate realistic, vivid-looking images without having any medical training data. Instead, synthetically generated data were used to train the NN.

In conclusion, the developed methods in this thesis can overcome the lack of samples and datasets.

In the future, the introduced methods can be used to design even better artificial intelligence-based medical support systems that can further assist physicians in clinical routines, diagnosis, therapy, or image-guided interventions, which can reduce clinical workload and thus improve patient safety.

Kurzfassung

Neuronale Netze haben in den letzten Jahren erstaunliche Ergebnisse bei der Erkennung von Ereignissen im Bereich der medizinischen Bild- und Videoanalyse erzielt. Dabei stellte sich jedoch immer wieder heraus, dass ein genereller Mangel an Daten besteht. Dieser Mangel bezieht sich nicht nur auf die Anzahl an verfügbaren Datensätzen, sondern auch auf die Anzahl an individuellen Stichproben, das heißt an unabhängigen Bildern und Videos, in bestehenden Datensätzen. Das führt wiederum zu einer schlechteren Erkennungsgenauigkeit von Ereignissen durch das neuronale Netz. Gerade im medizinischen Bereich ist es nicht einfach möglich die Datensätze zu erweitern oder neue Datensätze zu erfassen. Die Gründe hierfür sind vielfältig. Einerseits können rechtliche Belange die Datenveröffentlichung verhindern. Andererseits kann es sein, dass eine Krankheit nur sehr selten Auftritt und sich so keine Gelegenheit bietet die Daten zu erfassen. Ein zusätzliches Problem ist, dass es sich bei den Daten meist um eine sehr spezifische Domäne handelt, wodurch die Daten meist nur von Experten annotiert werden können. Die Annotation ist aber zeitaufwendig und somit teuer. Existierende Datenaugmentierungsmethoden können oft nur sinnvoll auf Bilddaten angewendet werden und erzeugen z.B. bei Videos nicht ausreichend zeitlich unabhängige Daten. Deswegen ist es notwendig, dass neue Methoden entwickelt werden, mit denen im Nachhinein auch Videodatensätze erweitert oder auch synthetische Daten generiert werden können.

Im Rahmen dieser Dissertation werden zwei neu entwickelte Methoden vorgestellt und beispielhaft auf drei medizinische Beispiele aus dem Bereich der Chirurgie angewendet. Die erste Methode ist die sogenannte Workflow-Augmentierungsmethode, mit deren Hilfe semantischen Information, z.B. Ereignissen eines chirurgischen Arbeitsablaufs, in einem Video augmentiert werden können. Die Methode ermöglicht zusätzlich auch eine Balancierung zum Beispiel von chirurgischen Phasen oder chirurgischen Instrumenten, die im Videodatensatz vorkommen. Bei der Anwendung der Methode auf die zwei verschiedenen Datensätzen, von Kataraktoperationen und laparoskopischen Cholezystektomieoperationen, konnte die Leistungsfähigkeit der Methode gezeigt werden. Dabei wurde Genauigkeit der Instrumentenerkennung bei der Kataraktoperation durch ein Neuronales Netz während Kataraktoperation um 2,8 % auf 93,5 % im Vergleich zu etablierten Methoden gesteigert. Bei der chirurgischen Phasenerkennung im Fall bei der Cholezystektomie konnte sogar eine Steigerung der Genauigkeit um 8,7 % auf 96,96 % im Verglich zu einer früheren Studie erreicht werden. Beide Studien zeigen eindrucksvoll das Potential der Workflow-Augmentierungsmethode.

Die zweite vorgestellte Methode basiert auf einem erzeugenden gegnerischen Netzwerk (engl. generative adversarial network (GAN)). Dieser Ansatz ist sehr vielversprechend, wenn nur sehr wenige Daten oder Datensätze vorhanden sind. Dabei werden mit Hilfe eines neuronalen Netzes neue fotorealistische Bilder generiert. Im Rahmen dieser Dissertation wird ein sogenanntes zyklisches erzeugendes gegnerisches Netzwerk (engl. cycle generative adversarial network (CycleGAN)) verwendet. CycleGANs führen meiste eine Bild zu Bild Transformation durch. Zusätzlich ist es möglich weitere Bedingungen an die Transformation zu knüpfen. Das CycleGAN wurde im dritten Beispiel dazu verwendet, ein Passbild von einem Patienten nach einem Kranio-Maxillofazialen chirurgischen Korrektur, mit Hilfe eines präoperativen Porträtfotos und der operativen 3D Planungsmaske, zu schätzen. Dabei konnten realistisch, lebendig aussehende Bilder generiert werden, ohne dass für das Training des GANs medizinische Daten verwendeten wurden. Stattdessen wurden für das Training synthetisch erzeugte Daten verwendet.

Abschließend lässt sich sagen, dass die in dieser Arbeit entwickelten Methoden in der Lage sind, den Mangel an Stichproben und Datensätzen teilweise zu überwinden und dadurch eine bessere Erkennungsleistung von neuronalen Netzen erreicht werden konnte.

Die entwickelten Methoden können in Zukunft dazu verwendet werden, bessere medizinische Unterstützungssysteme basierende auf künstlicher Intelligenz zu entwerfen, die den Arzt in der klinischen Routine weiter unterstützen, z.B. bei der Diagnose, der Therapie oder bei bildgesteuerten Eingriffen, was zu einer Verringerung der klinischen Arbeitsbelastung und damit zu einer Verbesserung der Patientensicherheit führt.

Acknowledgments

This thesis is the partial result of my scientific work as a research assistant at the Institute of Biomedical Engineering of the Karlsruhe Institute of Technology between May 2016 and November 2021. I want to express my gratitude to all those who made this work possible, supported me during this time, and thus contributed to the success of this thesis.

First, I would like to thank Prof. Dr. rer. nat. Werner Nahm for letting me join his research group, for his scientific enthusiasm, and his support in all matters until the completion of this work.
Further, I would like to thank PD Dr. Ing. habil. Thomas Wittenberg for his friendly interest in this thesis and for agreeing to act as a second examiner.

Moreover, I would like to thank Dr. med. dent. Reinald Kühle for providing me with important clinical data for my research. Without the data, a large part of this work would not have been possible, and for always being such a friendly and helpful person.

All my colleagues and former colleagues at the institute deserve a special thank you. A big thanks go to the persons who read parts of this thesis, and gave me feedback: Matthias Schaufelberger, Ady Naber, Nicolas Pilia, Ekatarina Kovacheva, Jorge Sánchez, Deborah Nairn, and Steffen Schuler.
Further special thanks go to many colleagues: Ady Naber for being a great companion, friend, listener over the whole time at the institute, Steffen Schuler for the incredible amount of scientific and non-scientific discussions during one or another beer, Ekatarina Kovacheva for the fun we had together during the four years she was my office mate, Nicolas Pilia for the special support during the completion of the dissertation, and Stefan Pollnow for the guidance at the beginning of my Ph.D. at the institute.

I want to thank all my students who have contributed to my project. Especially: Robin Andlauer, Jan Kost, Adrian Mohra and Christian Marzi.

A very special thanks go to the scuba divers of the TSG, who are always available for any fun and so make it possible to forget the worries and troubles for a short time.

Most of all, I would like to thank my family: my parents, Halina and Hermann, for their unconditional love and support during my study and all along the time, and for forming me into the person I'm now. To my sisters, Anette and Heidi, I would not be complete without them.

Finally, I would like to express my most profound gratitude to my friend and life partner, Heike, for always being and staying by my side in this adventure called life, motivating me in the most challenging times, and supporting me unconditionally. For this and much more, I love you!

Contents

Abbreviations

NN	neural network
CNN	convolutional neural network
RNN	recurrent neural network
LSTM	long short-term memory
GAN	generative adversarial network
WGAN	Wasserstein generative adversarial network
cGAN	conditional generative adversarial network
SGD	stochastic gradient descent
ADAM	adaptive moment estimation
HMM	hidden Markov model
DTW	dynamic time warping
RFID	radio-frequency identification
CBCT	cone-beam computed tomography
PNCC	projected normalized coordinate code
SSIM	structural similarity index
NME	normalized mean error
CDF	cumulative distribution function
AUC	area under the curve
HHMM	hierarchical hidden Markov model
ACC	accuracy
CBA	class balanced accuracy
AvACC	mean accuracy
$\mathbf{REC}_M$	macro-mean recall
$\mathbf{PREC}_M$	macro-mean precision
SPEC	specificity
$\mathbf{F1\text{-}score}_M$	macro F1-score

REC	recall
PREC	precision
pp	percentage points
CycleGAN	cycle generative adversarial network

Pre-publications of this Thesis

Parts of this thesis have already been published, verbatim or correspondingly, before submitting this thesis. The respective chapters are assigned to the corresponding references of the thesis in the Table below.

Chapter	Journal
Chapter 4	**Andreas Wachter**, and Werner Nahm. *Workflow Augmentation of Video Data for Event Recognition with Time-Sensitive Neural Networks*. preprint arXiv:2109.15063
Chapter 6	Robin Andlauer*, **Andreas Wachter***, Matthias Schaufelberger, Frederic Weichel, Reinald Kühle, Christian Freudlsperger, and Werner Nahm. *3D-Guided Face Manipulation of 2D Images for the Prediction of Post-Operative Outcome After Cranio-Maxillofacial Surgery*. IEEE Transactions on Image Processing vol. 30, pp. 7349–7363, Jan. 2021, doi:10.1109/TIP.2021.3096081.

Chapter 1

Introduction

1.1 Motivation

In recent years, the number of applications based on deep learning methods on image and video analysis in the medical field has continuously increased [1, 2]. The number of publications has also increased considerably, as shown in Figure 1.1. The application domains of neural networks (NNs) are diverse. They are used in image and video analysis, e.g., for methodical tasks like segmentation, recognition of surgical tools or classification and registration of lesions or cancer. Thereby, they support physicians during clinical interventions [3].

In the domain of image analysis and video analysis, outstanding results [1, 4–13] could be achieved by deep learning NNs. Isensee et al. [4] showed in the field of image segmentation that it is possible to segment ten different organs in computed tomography scans and magnetic resonance images [14] with only one NN instead of ten separate ones as it was customary before. In image recognition, Esteva et al. [5] showed that deep NNs provide a similar diagnosis as dermatologists at identifying skin cancer, and in some cases, NN was better than humans providing the diagnosis.

The application of supervised learning and NNs are often limited by the lack of annotated data [1, 3, 15–18]. The lack consists of an adequate amount of samples and datasets (as shown in Figure 1.2). Furthermore, datasets often suffer from balanced samples, diversity of diseases, manifestation, gender, age, ethnicity, etc. [1, 3, 15].

The community has widely recognized the lack of annotated data. Recently, datasets are becoming larger in quantity and diversity [13], e.g., the ChestX-ray8 [19] dataset consists of approximately 110,000 X-ray images, or the DeepLesion [20] dataset consists of 10,600 body-stem computed tomography scans from 4400 patients.

Nevertheless, these are all image datasets. For video datasets, the number of records is still very limited. As two examples for video datasets, the Cholec80 dataset [21] contains only 80 videos of different laparoscopic cholecystectomies or the Cataract dataset [22], consisting of only 50 different cataract surgical videos. This is countered by the generally

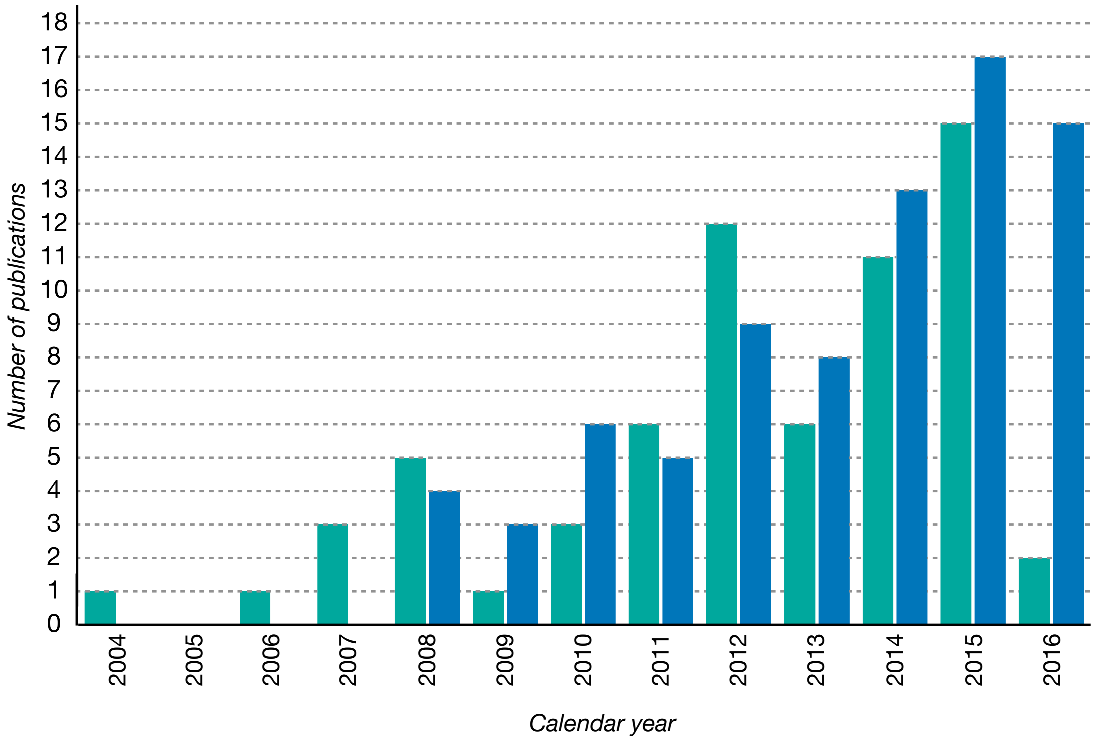


Figure 1.1: Number of publications in the period from 2004 -2016 in the field of medical image and video analysis, adapted from [1, 2]

accepted fact that deep learning is very data hungry [23].
Small datasets have an additional drawback. Since usually the annotation of the data is performed manually, the impact by inadvertently mislabeling in small datasets is larger than in large datasets [1]. Mislabeled samples can harm the performance, especially if only a few samples per class are available. These two issues, low amount of samples and potentially wrong labels, represent the central problem of insufficient data.

Due to the growing medical imaging community, the number of datasets and samples is expected to increase substantially in the next few years [24]. Nevertheless, it is possible that only a limited number of records are available due to the prevalence or manifestation of a specific disease. Additionally, the annotation has to be done by experts, i.e., physicians, which is very time-consuming and expensive. Therefore, expanding existing annotated datasets by newly developed techniques is desired [25].

Nowadays, image augmentation is a standard technique in computer vision to extend datasets artificially. Esteva et al. [5] or Al Hajj et al. [15] showed that image augmentation is an effective technique for object recognition or classification. Simple functions can generate a wide variety of images. Thus, simple functions like image shifting, cropping, rotating, or scaling, can significantly improve the generalization of an NN. Image augmentation works well in the medical image analysis domain for datasets with sufficient diversity. However, these preconditions are often not fulfilled, and mainly only common

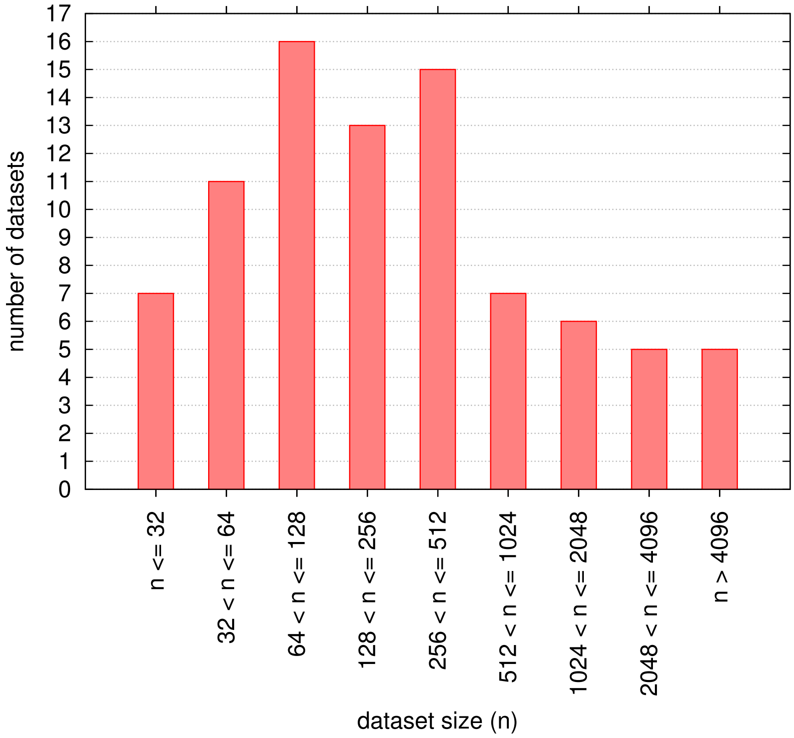


Figure 1.2: Distribution of dataset sizes in medical image and video analysis, adapted from [1]

disease patterns are included in the datasets. Though, physicians may need the most artificial intelligence support for rare diseases, because here they are in a situation beyond their everyday work [26].

Nevertheless, the improvements of using image augmentation in video analysis are limited because, an essential part of the information is usually coded in the variation of the frames over time or the events over time. Here, a model-based, or a generative adversarial networks (GANs)-based approach is auspicious and can remedy the lack of adequate data [27, 28].

Two of those approaches are developed and investigated in this work to enhance medical datasets artificially. The thesis contributes to the state-of-the-art model-based video augmentation and postoperative facial image estimation. Therefore, a method for the semantical augmentation of videos, and a method for training a conditional generative adversarial network (cGAN) on only synthetic training data, without including specific patient data, were developed to solve the technical need. The two introduced methods were finally evaluated on three surgical examples. The detailed objectives are formulated in the next section.

1.2 Objectives of the Thesis

The main focus of this work is set on the development and the usage of methods for the generation of artificial image and video data to enhance medical datasets for deep learning applications. As mentioned above, two different approaches are introduced in the thesis:

- A novel approach, that aims to solve the lack of suitable data in video datasets. For this purpose, a model-based augmentation method for video datasets has been developed. The method, called workflow augmentation, uses the respective semantic information, i.e., the workflow in the videos of a dataset, to generate retrospectively new artificial videos.
- A GAN-based approach that uses a patient-specific modifying 3D face model, e.g., from the surgery planning, and an arbitrary facial photo to generate a new photo with the face shape from the modified 3D model.

These two different approaches have been applied on three projects, each one with a specific goal:

The goal of the first project, with the title: *Workflow Augmentation of Video Data for Event Recognition with Time-Sensitive Neural Networks*, was to extend retrospectively and balance the training dataset from an existing cataract video dataset. Therefore, we use the workflow augmentation method to balance the frequency and timing of various unique events.
The hypothesis of this project was that balancing and augmentation of events, i.e., surgical tools, that rarely occur in the original dataset would lead to a better recognition performance of the NN. Therefore, the NN was trained with the augmented training dataset and compared the recognition performance with the same NN trained on the dataset using only image augmentation methods.

The goal of the second project, with the title: *Improving Surgical Phase Recognition in Videos using Workflow Augmentation*, was to augment and balance the semantic information with the workflow augmentation method. Hereby, the semantic information are the surgical phases in the training dataset on the example of cholecystectomy. Such semantic information cannot be augmented with state-of-the-art image augmentation methods because the information is not present in a single frame, instead it is encoded in the frame sequence that is not be manipulated by any image augmentation methods. Besides augmenting the dataset, another objective of the augmentation was to vary the lengths of the individual phase and balance the number of individual phase transitions. The idea was to avoid a biasing concerning the existing distribution in the original dataset. The project hypothesized that the classification accuracy of the individual phases can be improved by workflow augmentation compared to the literature.

The goal of the third project, with the title: *3D-Guided Face Manipulation of 2D Images for the Prediction of Post-Operative Outcome after Cranio-Maxillofacial Surgery*, was to find a suitable input parameter representation for cGAN of a modified 3D model. Since there was no suitable method or cGAN that could handle such an input vector so far. A further goal was to find a methodology for training a cGAN without having approximately enough pre- and post-operative image pairs of patients for supervised learning. The research question of this project was: how to overcome the lack of training data for training a cGAN with artificial, synthetically generated training data to estimate realistic post-operative patient-specific facial images that appear vivid and natural.

1.3 Structure of the Thesis

Part I introduces the relevant technical and medical fundamentals for understanding the presented approaches and results:

- **Chapter 2** provides the technical fundamentals of neuronal networks.
- **Chapter 3** introduces the three different surgical procedures that serve as examples of the two developed approaches.

Part II presents the three different projects that were worked on as part of the thesis.

- **Chapter 4** presents the first project entitled *Workflow Augmentation of Video Data for Event Recognition with Time-Sensitive Neural Networks.* For this project, a specific introduction is given with the goals of the research project and the hypothesis. Then, the state-of-the-art in the field of video augmentation is presented. After that, the developed workflow augmentation approach is described in detail. Subsequently, the results are presented and compared to one state-of-the-art method using the example of a cataract surgery. Finally, the results are discussed regarding the hypotheses, and a conclusion is given.
- **Chapter 5** is about the second project with the title *Improving Surgical Phase Recognition in Videos using Workflow Augmentation.* An introduction to the state-of-the-art surgical phase recognition using neural networks is given. The hypothesis of the project is then formulated. Afterwards, the parameters for workflow augmentation are defined. Subsequently, the augmented dataset is analyzed, and the phase detection on cholecystectomy as an example is presented. Finally, the results are classified and evaluated concerning the literature, and a conclusion is given.
- **Chapter 6** covers the third project titled: *3D-Guided Face Manipulation of 2D Images for the Prediction of Post-Operative Outcome after Cranio-Maxillofacial Surgery.* First, an introduction to cranio-maxillofacial surgery planning and state-of-the-art facial manipulation approaches will be given, and then the research gap and question will be identified. Subsequently, the procedure will be described with developed methods and the training of the NN. After that, the qualitative and quantitative results of the approach will be demonstrated with the use of celebrity

portraits. Afterwards, the experiment results with real patient photos and 3D models are shown. Finally, the results are discussed and summarized.

Part III summarizes and discusses the results of all projects regarding the developed approaches from a general perspective and gives a conclusion. Furthermore, possible future research topics are identified.

- **Chapter 7** presents the general restriction of the developed approaches and the general conclusion.
- **Chapter 8** gives an outlook on possible future work based on the thesis results.

PART I

FUNDAMENTALS

Chapter 2

Artificial Neural Network

Artificial neural networks (NNs) are a powerful approach for data analysis. Thereby, the types of input data are not restricted. These can be, e.g., discrete 2D waveform signals, images, or videos. The basic principles and different types of NNs will be explained in the following.

2.1 Artificial Neuron

The artificial neuron is the fundamental component of NNs. Figure 2.1 shows the principal components of an artificial neuron. It bases on the behavior observed in biological neurons. Each neuron has at least one input and a single output. The output value is determined by an input function $\varphi(\mathbf{x}, \mathbf{w})$ dependent on the input values $\mathbf{x}$ and the weights $\mathbf{w}$, usually with a constant bias term $\mathbf{w_0}$. The function $f(\varphi)$ is the respective activation function of the neuron. Neurons can be cascaded, so the output value y can be "sent" to other neurons and can be used as an input value. The user can arbitrarily select the concrete input function and activation function, see [29].

2.2 Perceptron

A perceptron is a network of connected neurons organized in layers. A perceptron has at least an input and an output layer. The perceptron is characterized by linear activation functions in the neurons, except the input node. The input neuron has only the functionality to store the input values. The number of neurons of the input and output layers corresponds to the number of input and output variables, respectively. Furthermore, it is also possible to have several layers located in between. This layers are called hidden layers. A perceptron consisting of at least one hidden layer is called multi-layer perceptron. Each hidden layer consists of a predefined number of neurons. In addition, the different layers are interconnected.

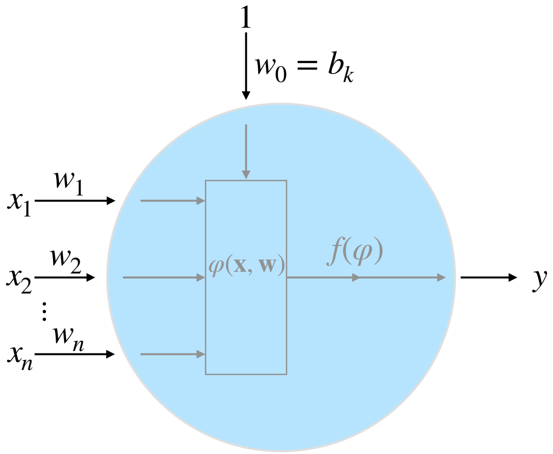


Figure 2.1: Concept of an artificial neuron model; $\mathbf{x}$ are the input values; $\mathbf{w}$ are the weights per input value; w_0 corresponds for the bias b_k; $\varphi(\mathbf{x}, \mathbf{w})$ is the summing input function; $f(\varphi)$ is the activation function; y is the output of the neuron.

The way the layers are connected is called topology, shown in Figure 2.2. A layer where every neuron of the previous layer is connected to every neuron of the succeeding layer is called a fully connected layer, but not all neurons within one layer must be connected to a neuron of the next layer. Besides, there can also be reversed or loop interconnection between neurons of the same or different layers. [30]

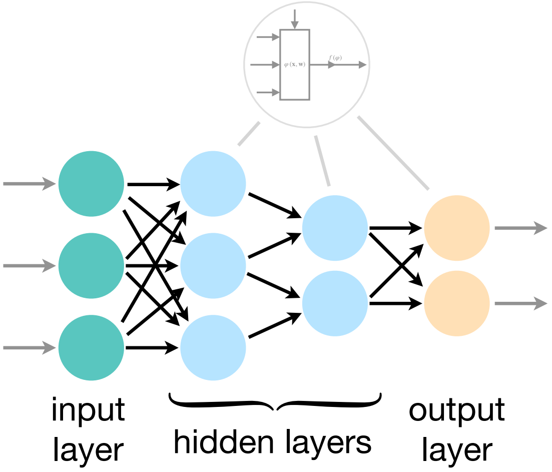


Figure 2.2: Example of a three-layer perceptron with two hidden layers (blue), an input (green), and an output layer (orange). The input layer and the first hidden layer are fully connected. Only forward connections are used, and the preceding and succeeding layers are connected.

2.3 Convolutional Neural Networks

A convolutional neural network (CNN) is one derivative of the multi-layer perceptron. Over time, CNNs replaced the multi-layer perceptrons, which were formerly used in computer vision, for various reasons. For example, only linear problems can be separated with a single layer perceptron. Multiple hidden layers were needed for more complex tasks encountered in modern advanced computer vision or medical image analysis. In the case of using a perceptron, the number of weights increases very fast due to the fully connected layers. Additionally, the spatial information in an image would also not be considered, which can only cover a small region of the entire image.

CNNs use flattening functions, so-called convolutional kernels, as shown in Figure 2.3, to take the spatial information and relation into account. With the help of kernels, the CNN tries to extract semantic features, e.g., edges, homogeneous faces and patterns [31]. For this purpose, a series of convolution operations (with different kernels) are applied to the image. Here, the convolutional kernels usually grasp only small parts of the image. In principle, the used kernels' number, size, and shape are not limited. The results of the convolutions form a series of outputs, and these are called feature maps. At the same time, some operations allow reducing the number of variables. The reduction is made by so-called pooling layers. This reduction can be repeated as often as needed just by inserting these pooling layers into the NN structure, as shown in Figure 2.3 (3). Finally, the outputs are fed into at least one fully connected layer that calculates the regression output of the CNN, shown in Figure 2.3 (4). The convolution operation and the down-sampling allow deeper neural networks consisting of a higher number of hidden layers. Such a type of NNs is called deep NNs. Furthermore, it should be pointed out that CNNs mostly contain only forward-directed connections, which allows using and extracting only information from the current input.

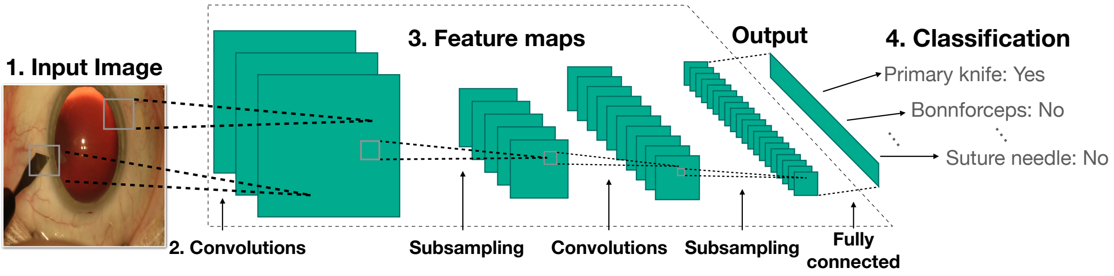


Figure 2.3: Illustrates the schematic procedure of a CNN. 1. is the entire input image. In 2. single regions from the image are extracted by kernel functions of the CNN. In 3. feature maps are calculated by the CNN to output a classification vector for the input image in the final step 4.

2.4 Recurrent Neural Networks

To consider the temporally encoded information or features from previous inputs, the NN structure is extended by reverse connections to a preceding layer or feedback loops to the current layer. Such a class of NNs is called recurrent neural networks (RNNs). RNNs allow extracting information also from the input history. However, the time interval in which the already seen information is considered is restricted. The limit results from the fact that the influence of the current input decreases with each step, i.e., information that lies longer in the past is taken into account less and less and at the same time leads to a vanishing gradient. This problem is also known as the vanishing gradient problem. Thereby, during the training of the NN, the update gradient is so tiny that the input weight of the neuron barely or does not change its value. These tiny weight updates lead to the situation that adjusting the weights either takes a very long time or the network remains in one state [32]. The long short-term memory (LSTM) network tries to overcome the problem of a long training time [33]. The concept will be presented in the following section.

2.4.1 Long Short-Term Memory

A LSTM-block consists of three different gates and a central memory cell, see Figure 2.4. The unique memory cell c_t stores the value using a feedback loop. The three gates are the input gate i_t, output gate o_t, and forget gate f_t. The input gate controls the extent of the influence of a new value that flows into the memory cell at a time step t. The forget gate determines the degree of a stored value stays in the cell or is forgotten. The output gate controls whether the stored value is used to calculate the next module of the network or not. For the interaction of the different components, $tanh$ functions,

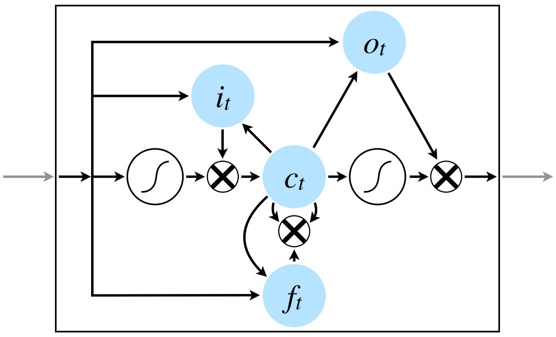


Figure 2.4: Long short-term memory block with $tanh$ activation function ∫ and convolution operations ⊗. The memory cell c_t stores the value. The input gate i_t, that controls the extent a new value. The output gate o_t controls whether the stored value is output. The forget gate f_t determines the degree to which a stored value stays of would forgot. Adapted from [34], licensed under Creative Commons Attribution-Share Alike 4.0 International.

various vectors, and matrix operations are used. Instead of the fixed weights (w_i), the product of weights and the associated gate is used. This structure tries to imitate the natural behavior of memorization, which includes forgetting and memorization. The LSTM-block allows keeping the information in the network for long periods, but also to have an adaptive influence the impact on a current time point. [31, 35]

2.5 Generative Adversarial Networks

All the previously described network topologies solve discriminative tasks, but NN can also perform well for generative tasks. Therefore, generative models like generative adversarial networks (GANs) [36] are used. Thereby, GANs aim to generate samples that follow real data distribution. GANs have shown remarkable results in image synthesis, image translation, and other generative tasks in computer vision. GANs consist of two neural networks, as shown in Figure 2.5. One part of the network is the generator $\mathbf{G}$, the second is the discriminator $\mathbf{D}$. The generator produces the data, and the discriminator judges them. The generator tries to mimic the real data distribution by generating new images using 2D images containing, e.g., white Gaussian noise. The discriminator then rates with which probability $\mathbf{P}$ the result of the generator belongs to the real $\mathbf{P_r}$ or fake $\mathbf{P_f}$ distribution. Hence, the generator aims to learn generating results according to the distribution of the real data, and the discriminator aims to distinguish the generator's results from the real data. This objective was formulated as a minimax objective by Goodfellow et al. [36]:

$$\min_{G} \max_{D} V(\mathbf{D}, \mathbf{G}) = \mathbb{E}_{x \sim \mathbf{P}_r}[\log \mathbf{D}(x)] + \mathbb{E}_{x \sim \mathbf{P}_f}[\log(1 - \mathbf{D}(x))] \tag{2.1}$$

Here, x is the input image, and $\mathbb{E}_{x \sim \mathbf{P}_r}$ is the expected value that the image belongs to the set of real images and $\mathbb{E}_{x \sim \mathbf{P}_f}[\log(1 - \mathbf{D}(x))]$ is the expected value that the image belongs to the set of fake images. Thus, after successful training, the generator can output images indistinguishable from real images.

Both NNs, $\mathbf{G}$ and $\mathbf{D}$, were trained in an alternating manner. The update can result in a zero-sum game, where each try to "win" the adversarial game. The main problem of GANs is the loss of convergence during training [37], especially if the initial distributions of real and fake images are too far from each other. This problem also correlates to the vanishing gradients problem [32]. Both networks can end up in a deadlock, and the generator produces only a limited number of sufficiently different samples, or they end within oscillating results. In the following sections, one approach for overcoming these training problems of GANs are presented.

2.5.1 Wasserstein Generative Adversarial Networks

To overcome the main problem of GANs, many proposals have been made in recent years. One proposal is the Wasserstein generative adversarial network (WGAN) by

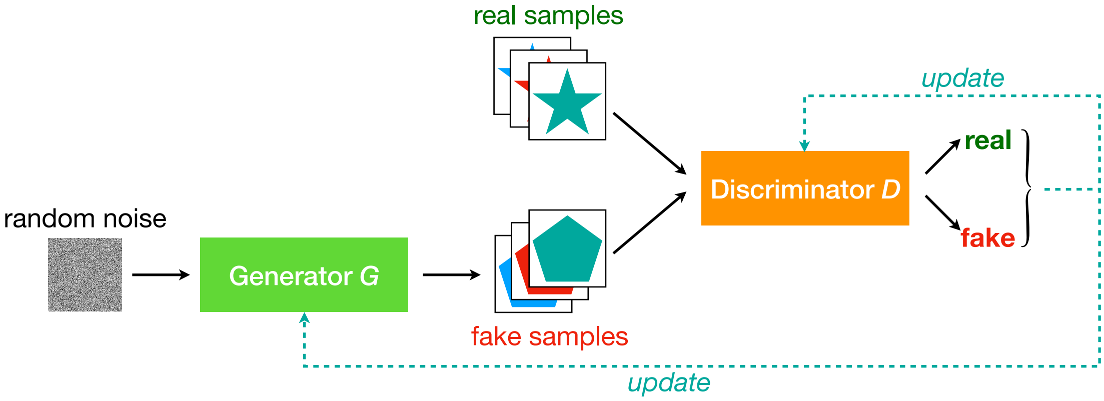


Figure 2.5: Setup of a GAN which was proposed by Goodfellow et al. [36]. The generator tries to mimic the real data distribution by generating new images out of latent 2D noise vectors. The discriminator then rates with which probability the result of the generator belongs to the real or fake distribution.

Arjovsky et al. [38], which is based on the Wasserstein distance, also known as the earth-mover distance [39]:

$$W(\mathbf{P}_r, \mathbf{P}_f) = \inf_{\gamma \in \Pi(\mathbf{P}_r, \mathbf{P}_f)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}[||\mathbf{x} - \mathbf{y}||] \tag{2.2}$$

The Wasserstein distance $W(\mathbf{P}_r, \mathbf{P}_f)$ is the minimum cost of moving and transforming a mass, such as a pile earth $\mathbf{x}$ corresponding to data distribution $\mathbf{P}_f$ into a pile $\mathbf{y}$ of a different shape, corresponding to data distribution $\mathbf{P}_r$. The Wasserstein distance for the real data distribution $\mathbf{P}_r$ and the generated data distribution $\mathbf{P}_f$ is mathematically defined as the largest lower bound of all transformation cost $\Pi(\mathbf{P}_r, \mathbf{P}_f)$ (i.e., amount of earth moved times the movement distance). The advantage of using a cost function based on Wasserstein distance is that it has a more linear gradient, as shown in Figure 2.6. The Figure shows a distribution plot for the value of $\mathbf{D}(X)$ for GAN and WGAN. For the GAN, red line, in the ranges of high and low values for D, the gradients are close to 0 or 1. As a result, the update changes are very large or negligible during the training. For WGAN (the blue line), gradient slope is always unequal or not close to zero, which facilitates successful training.

2.6 Conditional Generative Adversarial Networks

An extension of this concept of GAN is the conditional generative adversarial network (cGAN). In addition to the generated sample being realistic, an additional condition c must be fulfilled. cGANs are used, for example, for image-to-image translation. Here an image from one domain $\mathbb{A}$ is transferred into an image from another domain $\mathbb{B}$. The domain refers to a certain characteristic of a set of images. However, corresponding cross-domain image pairs $(x_{\mathbb{A}}, x_{\mathbb{B}})$ must be available for the training of a cGAN.

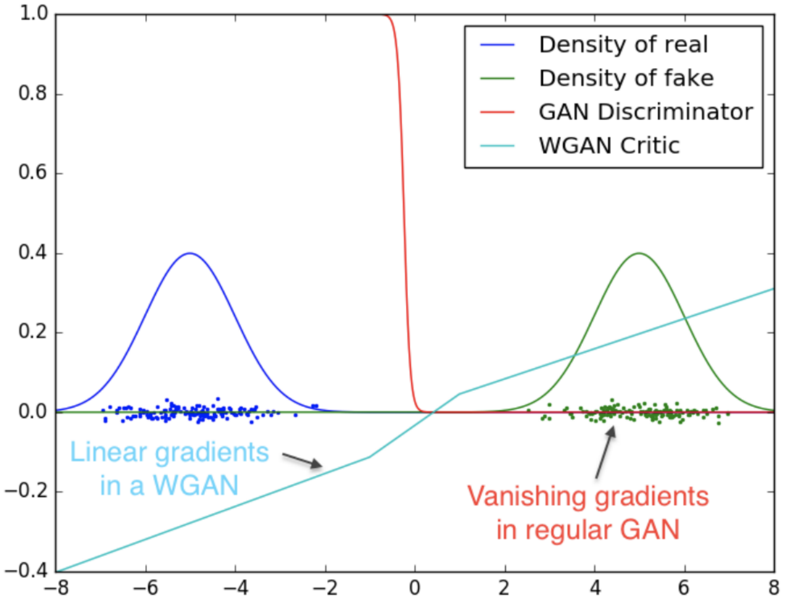


Figure 2.6: Comparison of the Wasserstein generative adversarial network discriminator and the GAN discriminator. The discriminator of the GAN has to be distinguished between real samples (dark blue) and fake samples (green). After convergence, the red line shows the prediction across the x-axis where the samples were located. As it can be seen, the red line gradients were close to zero at the location of the fake samples, which hindered the training of the generator **G**. On the other hand, the metric of the Wasserstein generative adversarial network (light blue) provided "sufficiently nice" [38] gradients between the distributions. Note that Arjovsky et al. did not describe the axes. Image adapted from [38]

.

To overcome this problem, Zhu et al. introduced cycle generative adversarial network (CycleGAN), which provides cross-domain image translation without requiring corresponding image pairs for training [40]. For updating the weights of the CycleGAN, the predicted image is not compared with the ground-truth image in domain $\mathbb{B}$. Instead, the predicted image is transposed back to the original domain $\mathbb{A}$ using the same NN and compared against the original image.

Nevertheless, the CycleGAN has one major drawback: it must train $k(k-1)$ generators $\mathbf{G}$ to translate bidirectionally between k domains. For example, if someone wants to translate between images of people with brown, blond, and black hair, six generators $\mathbf{G}$ and three discriminators $\mathbf{D}$ must be trained. To overcome this drawback again, Choi et al. [41] proposed a modified CycleGAN that relies on a single generator $\mathbf{G}$ and a single discriminator $\mathbf{D}$. In addition, a label c was defined to describe the domain transformations. Furthermore, the discriminator $\mathbf{D}$ is trained not only to discriminate between fake and real images but also to predict the label c of the given image. Then, the domain prediction of the discriminator $\mathbf{D}$ for a generated image was minimized by the generator $\mathbf{G}$ in addition to the GAN objectives. As a result, the cGAN approach allowed larger datasets for training, which led to a better generalization and image quality of the generator $\mathbf{G}$.

2.7 Training Neuronal Networks

Training a NN means that the weights of the connections between the neurons are optimized stepwise regarding an objective. In general, a distinction is made between supervised and unsupervised learning strategies. The network does not receive pre-assigned labels for the training data in unsupervised learning. Therefore, the network has to recognize patterns in the images or input data on its own, which can be very difficult depending on the domain. This kind of training is not used in this work and, for this reason, is not described in detail.

In contrast, the NN's output is mapped with the known label vector to the given input vector for supervised learning or training. Thereby, the transition weights $\mathbf{w}$ are incrementally adjusted according to the input label pair such that the output $\mathbf{y}$ of the network matches given labels $\mathbf{l}$ concerning given inputs $\mathbf{x}$ [31]. The labels were assigned manually before for the training data. Finally, supervised learning is an optimization problem in which an objective function is solved during training. The estimation error, which is defined by the loss function $\mathscr{L}$, should be minimal [42]. Then optimization problem that has to be solved during the training can described as:

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}}\, \mathscr{L}(f_{net}(\mathbf{w}, \mathbf{x}), \mathbf{l}) \tag{2.3}$$

Depending on the specific task, the loss function $\mathscr{L}$ is differently chosen. The optimization methods which were used in this work are described below.

2.7.1 Stochastic Gradient Descent

The stochastic gradient descent (SGD) is an extension of the method of gradient descent. Gradient descent is problematic when the NN has several million weights. The more unknown variables (i.e., weights), the larger the dataset needed to calculate the optimal solution. The gradient descent method searches for a locally optimal solution on the entire dataset. Thereby, the idea is to move along the function's gradient $\mathscr{L}$ towards the minimum, as shown with the black line in Figure 2.7 (a). The gradient describes the behavior of the network and is used to iteratively change $\mathbf{w}$ so that the loss function becomes minimal [42, 43], as shown in Figure 2.7 (b).

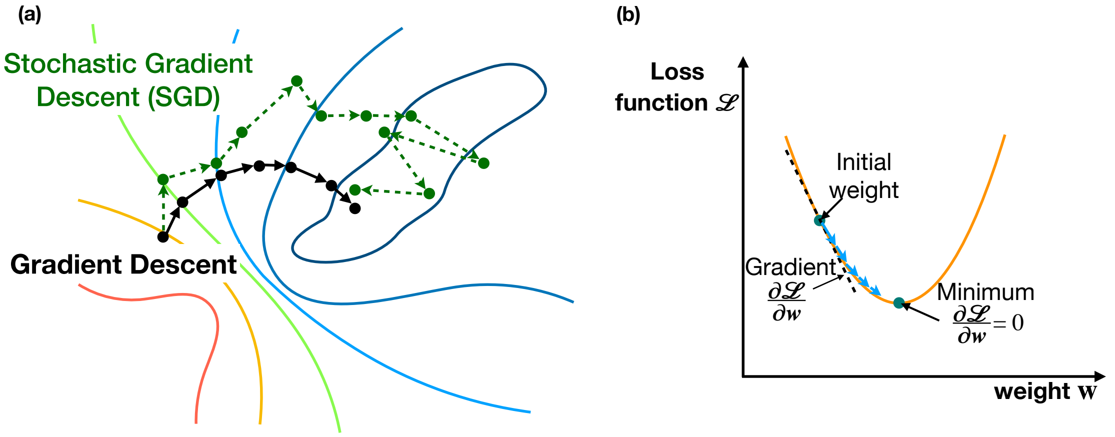

(a)
Stochastic Gradient Descent (SGD)
Gradient Descent
(b)
Loss function $\mathscr{L}$
Initial weight
Gradient $\frac{\partial \mathscr{L}}{\partial w}$
Minimum $\frac{\partial \mathscr{L}}{\partial w} = 0$
weight w

Figure 2.7: (a) Direct graphical comparison of the convergence steps to reach the minimum of the loss function of the gradient descent approach (in black) and the stochastic gradient descent (in green). (b) Schematic of the loss function convergence for the initial state to the minimum.

During NN training the weights $\mathbf{w}_j$ are updated as follows:

$$\mathbf{w}_{j+1} = \mathbf{w}_j - \delta \nabla_{\mathbf{w}} \mathscr{L}((f_{net}(\mathbf{w}_j, \mathbf{x}), \mathbf{l}) \tag{2.4}$$

It should be noted that an initial weight vector $\mathbf{w}_0$ or the step size, also known as the hyperparameter *learning rate* δ, must be given. Mostly a random initialization is made for the weight vector to avoid side effects like biasing.

As mentioned above, the gradient descent can fail for problems that are containing several million variables. However, the computational problem of finding the optimal solution for large networks can be solved by partial updating on a subset of the dataset. For this purpose, the entire dataset is divided randomly into subsets, and the gradient computation is solved on only a subset. It could be proven that the summed-up solution of the sub-problems corresponds to the gradient descent solution [42]. This technique is called stochastic gradient descent (SGD), and the strategy during the training for weight updates can be formulated as follows:

$$\mathbf{w}_{j+1} = \mathbf{w}_j - \delta \nabla_{\mathbf{w}} \mathscr{L}((f_{net}(\mathbf{w}_j, \mathbf{x}), \mathbf{l}) = \mathbf{w}_j - \frac{\delta}{N} \sum_{i}^{N} \nabla_{\mathbf{w}} \mathscr{L}((f_{net}(\mathbf{w}_{ji}, \mathbf{x}_i), \mathbf{l}_i) \tag{2.5}$$

Here N is the total number of label-input pairs. Furthermore, SGD takes advantage of the fact that the gradient can be approximated. For this purpose, a set $M < N$ is selected from the sub-problem. This subset, also-called mini-batch or batch, is then used to update all the mesh weights. The batch size, a further hyperparameter, has to be chosen. Furthermore, the practice has shown that for the SGD method, it is essential that the learning δ adaptively decreases as the loss function is converged. The reason for the decrease in the learning rate is that the random batch selection is comparable to a noise source because not all classifications are included in a batch. The random batch selection can lead to the loss still oscillating slightly when the optimum is reached [42, 44].

2.7.2 Adaptive Moment Estimation

The adaptive moment estimation (ADAM) method is a gradient-based algorithm optimizing stochastic objective functions [45]. Compared to the SGD algorithm ADAM does not have a fixed learning rate but it calculates it adaptively. For this purpose, an individual learning rate is assigned to each parameter at each update. The weights $\mathbf{w}$ are updated in the following way:

$$\mathbf{w}_{j+1} = \mathbf{w}_j - \delta \frac{\widehat{m}_{j+1}}{\sqrt{\widehat{v}_{j+1}} + \varepsilon} \tag{2.6}$$

Here, δ is the pre-defined learning rate and $\varepsilon = 10^{-8}$ being a constant for numerical stability. The first and second moments $\widehat{m}$ and $\widehat{v}$ are estimated by:

$$\widehat{\mathbf{m}}_{j+1} = \frac{\beta_1 \mathbf{m}_j + (1 - \beta_1)\mathbf{g}_{j+1}}{1 - \beta_1} \tag{2.7}$$

$$\widehat{\mathbf{v}}_{j+1} = \frac{\beta_2 \mathbf{v}_j + (1 - \beta_2)\mathbf{g}_{j+1}^2}{1 - \beta_2} \tag{2.8}$$

Here $\mathbf{g}_{j+1} = \nabla_{\mathbf{w}} \mathscr{L}((f_{net}(\mathbf{w}_j, \mathbf{x}), \mathbf{l})$ with the loss function $\mathscr{L}$. β_1 and β_2 have to be pre-chosen. For computer vision tasks, ADAM is one of the most popular optimizers besides SGD due to its fast convergence. However, it must also be noted that the SGD algorithm is significantly slower for some specific deep learning tasks but can find a better solution [46].

2.8 Back-propagation Algorithm

The back-propagation algorithm is applied to update the weights in the different layers in supervised learning [30]. Here, the chain rule is used to calculate the gradient for each weight. The chain rule states that the weight in one layer depends only on the weights and outputs of the subsequent layers. Thus, Equation (2.3) can be solved layer-wise, and the weights are iteratively updated from the output layer backward. The weight updating is always done after calculating the losses for a given batch. Therefore, the loss is usually averaged over all samples.

2.9 Regularization Methods

In machine learning, there are several regularization techniques to avoid overfitting. Overfitting means that a NN memorizes instead of recognizing the input-label pairs. Overfitting reduces the generalizability, i.e., the network's performance on new unknown data is considerably worse than on the training data. The transfer learning method is often used in the medical field, and the data augmentation method is usually used in image analysis. In practice, these methods are combined to improve the generalization of NN. The two methods will be shortly introduced, in the following subsections.

2.9.1 Transfer Learning

Transfer learning are used since there is usually not enough data available to train deep NN from scratch completely. Thereby, the NN uses the same characteristics in images on two different datasets [42]. For example, in medical image analysis, the network is trained on standard images, such as the ImageNet dataset [47], which contains dogs, flowers, cats, and furthermore. The expectation is that the network learns to recognize semantic content, like edges, monochromatic areas, patterns, and others. This semantic content could also be found in medical images or other images. Afterwards, the pre-trained network with the stored weights is taken and transferred to the application domain by continuing the training with the available data, e.g., medical images. By doing this, the required amount of data can be reduced, and overfitting can be avoided. For transfer learning, the output layer, which is domain-dependent, must usually be adapted in practice. Hence, a re-initialization of the weights in the output layer is necessary.

2.9.2 Data Augmentation

Data augmentation is another possibility, besides transfer learning to avoid overfitting on small datasets. Data augmentation is an artificial extension of an existing dataset. In practice, it is relatively easy for some machine learning tasks to create such data. For example, by simple spatial transformations (rotating, scaling, adding noise, etc.), a "new" image can be generated for, e.g., image classification using CNN. For a CNN, the transformed image is new because the local information is swapped, and therefore the input values look different for the CNN. As described in Section 2.3, a CNN tries to extract semantic information from the input image. Hence, an image is rotated, the semantic information in the image has to be shifted and therefore has to be extracted by other neurons. In practice, dataset augmentation techniques have been shown to decrease the generalization error of machine learning dramatically.

Chapter 3

Medical Fundamentals

In the following, the three different types of surgery are briefly described, which have been necessary for this work. They form examples for the applications of the artificial data generation approaches.

3.1 Cataract Surgery

With 19 million surgeries performed annually, cataract surgery is the most common surgical procedure globally [48]. Cataract describes the disease where the natural lens of the eye becomes cloudy. The reason for this is usually physiological aging of the crystalline lens, but there are also other factors such as metabolic disorders [49].
The clouding reduces the field of view. In remarkably progressed disease states, the lens clouding is so pronounced that the incident light can no longer be focused on the retina, see Figure 3.1. Furthermore, it can also lead to complete blindness. The only treatment option at the moment is the cataract surgery. Thereby, the most common procedure is extracapsular cataract extraction, here the cloudy lens in the capsular bag is replaced with an artificial lens implant.

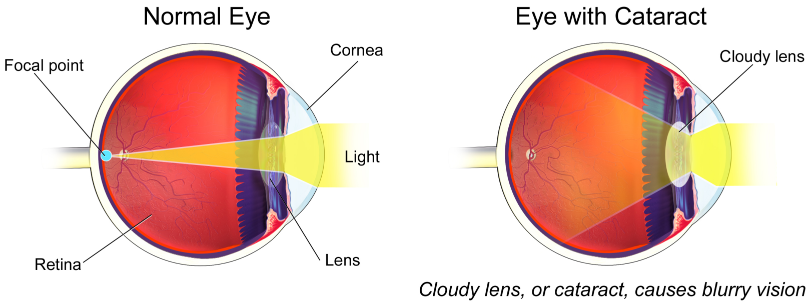


Figure 3.1: Schematic of the eye from a healthy crystalline lens (left) and cataract lens (right). Modified from [50], licensed under Creative Commons Attribution-Share Alike 4.0 International.

The operation is a minimally invasive surgical procedure that is highly standardized. The entire procedure can be performed with only small incisions in the cornea. The incisions are usually only about 2 mm in width. Afterwards, the capsular bag is opened, and the cloudy lens is removed. A femtosecond laser has been increasingly used to open the capsular bag and cornea in recent years. This procedure is less invasive and more precise [51].

Then the natural lens is usually removed by phacoemulsification, i.e., a high-frequency ultrasound device is used to disrupt the lens into small pieces and simultaneously they are aspirated. A so-called injector is used to insert the artificial lens. With its help, the flexible artificial lens is rolled up before being inserted into the eye and positioned in the capsular bag. The procedure usually takes 15 minutes. Immediately after the surgery, the patient recovers his visual capability [52].

3.2 Cholecystectomy Surgery

Gallbladder removal, also called cholecystectomy, is performed more than 750,000 times per year in the United States. Other studies says that 10– 15 % of adults are affected by gallstone disease in their lifetime [54]. The gallbladder is a hollow organ attached to the liver. The gallbladder stores the digestive fluid bile, a fluid produced by the liver that helps metabolize fat.

Removal of the gallbladder is usually necessary because of gallstones. Depending on their size of the gallstones, they can lead to a blockage of the gallbladder system. It results in an inflammation of the gallbladder (cholecystitis). The cholecystectomy is usually performed by minimally invasive surgery, or more precisely laparoscopically. In this treatment, a tiny video camera and special surgical instruments are inserted through

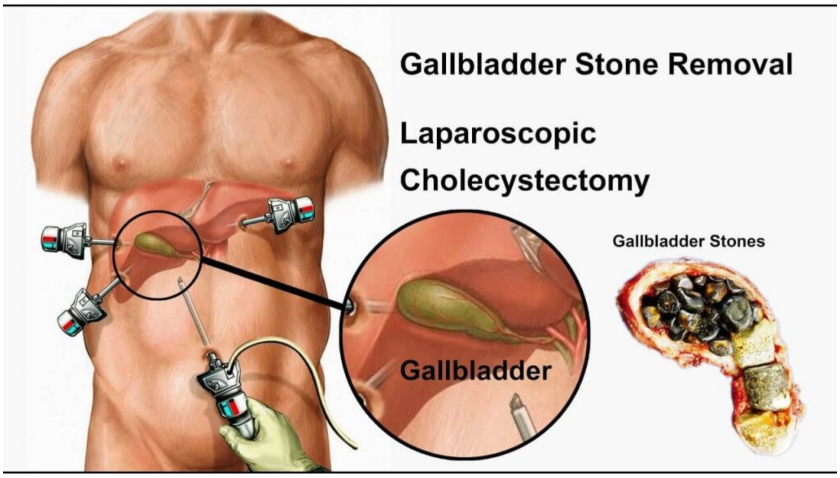


Figure 3.2: Schematic of a laparoscopic Cholecystectomy (left). A figure of a extracted gallbladder, which was filled with stones (right). Image from [53]

small incisions in the abdominal wall, as shown in Figure 3.2. The abdominal chamber is additionally inflated with carbon dioxide for better visibility and a larger working room. Laparoscopic cholecystectomy takes about one to two hours.

3.3 Cranio-maxillofacial Surgery

Cranio-maxillofacial surgery is a standard treatment for temporomandibular joint disorders or skeletal deformities, e.g., dysgnathia. Dysgnathia refers to the non-physiological development of the masticatory apparatus components to a nominal jaw position. The reason for these misalignments can be congenital or acquired. In the case of congenital dysgnathia, the reason is usually a delay or irregularity in the embryonic development of the skull. Acquired dysgnathia is usually the consequence of, e.g., trauma, tumors, or a malfunction of the swallowing, chewing, and tongue muscles. [55]

The consequence of dysgraphia can be tooth misalignment, occlusion disorders, and aesthetic and functional impairments, e.g., a crossbite, a cover bite, a scissor bite, or an open bite [56]. In addition to conservative dental and orthodontic treatment, surgery may be induced depending on the grade of this disease, e.g., in case of bone malformations. During the operation, the jaw will be adjusted. The operation is performed at an adult age so that bone growth does not lead to a new misalignment. Besides improving the function, the jaw correction according to the standard shown in Figure 3.3 [57, 58], can significantly change the patient's facial appearance. The decision for or against the surgery can be very burdensome for the patient.

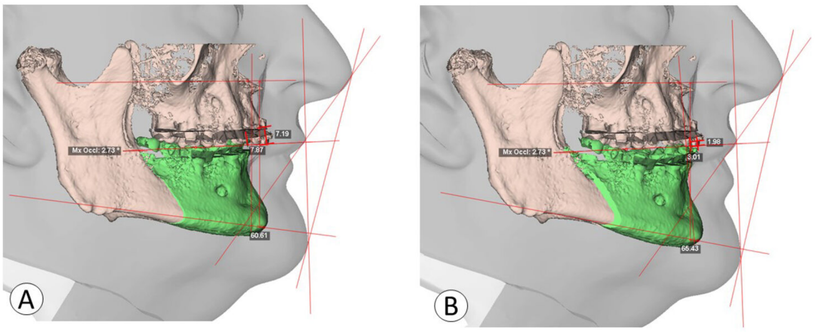


Figure 3.3: Planning a dysgnathia surgery in which an underbite (A) is to be corrected. The lower jaw is divided (green) and pushed forward slightly (B). [59]

PART II

PROJECTS

Chapter 4

Workflow Augmentation of Video Data for Event Recognition with Time-Sensitive Neural Networks

4.1 Introduction

In recent years, deep learning has shown excellent results in medical image analysis and event recognition. Deep learning is a promising method to support physicians in their diagnostic and clinical daily routine. Today, deep learning applications are already assisting physicians in diagnosis [5], image registration [10], multi-modal image analysis [11], and image segmentation [12]. The most common type of deep learning network is the convolutional neural network (CNN) [60]. CNN is engineered to extract information from an image using multiple convolutional kernels. However, CNNs have also been used for event recognition in surgical workflows from videos, in recent years [15, 61–64]. Event recognition is typically done by detecting surgical instruments in the images. Nevertheless, the CNN still considers only the image information of individual frames without including the information from the chronological sequence of events. However, in most cases, a surgical procedure usually follows a predetermined established workflow. Hereby, specific instruments appear in the characteristic phases of the workflow. As context recognition has increasingly come into focus in surgery [65, 66], recognizing the chronological sequence of events becomes very important. Morita et al. [67] used for the surgical phase recognition in cataract surgery the Inception V3 model, which is based on a CNN. Twinanda et al. [21] used the EndoNet, an extension of a CNN by a hierarchical hidden Markov model (HHMM), for detecting the individual phase of laparoscopy. The CNN extracts the image features, and the HHMM considers all temporal information. Twinanda et al. also mention that the HHMM is trained separately from the CNN, due to less training data. Al Hajj et al. [15] were able to demonstrate that, instead of a combination of CNN and HHMM, the combination of CNN and recurrent neural network (RNN), i.e., long short-term memory (LSTM), can considerably increase the recognition performance. However, both approaches from Al Hajj et al. [15] and Twinanda et al. [21] still suffer from small and highly unbalanced training datasets. The authors found out that the precision for detecting a specific tool is highly correlated with the prevalence of the tool in the training set. This dependence of recognition rate and frequency is another indicator of the main problem of many deep learning applications in medicine. Frequently, medical datasets are too small and unbalanced for effective training. The consequences are insufficient detection rates and a lack of robustness, especially for rare events.

Therefore, we conclude that the availability of sufficiently large and adequately balanced training datasets is a prerequisite for the use of deep neural networks (NNs). The conclusion leads directly to this paper's scientific question: How can existing surgical video datasets be retroactively enlarged and balanced, specifically regarding the frequency and the chronological sequence of events?

This study presents a novel end-to-end workflow-based approach for augmenting and balancing surgical videos. In contrast to previous approaches [15, 61, 68], we propose a combination of several methods whose starting point is the extraction of the surgical workflow. Then using this workflow, the artificial videos are reassembled and augmented

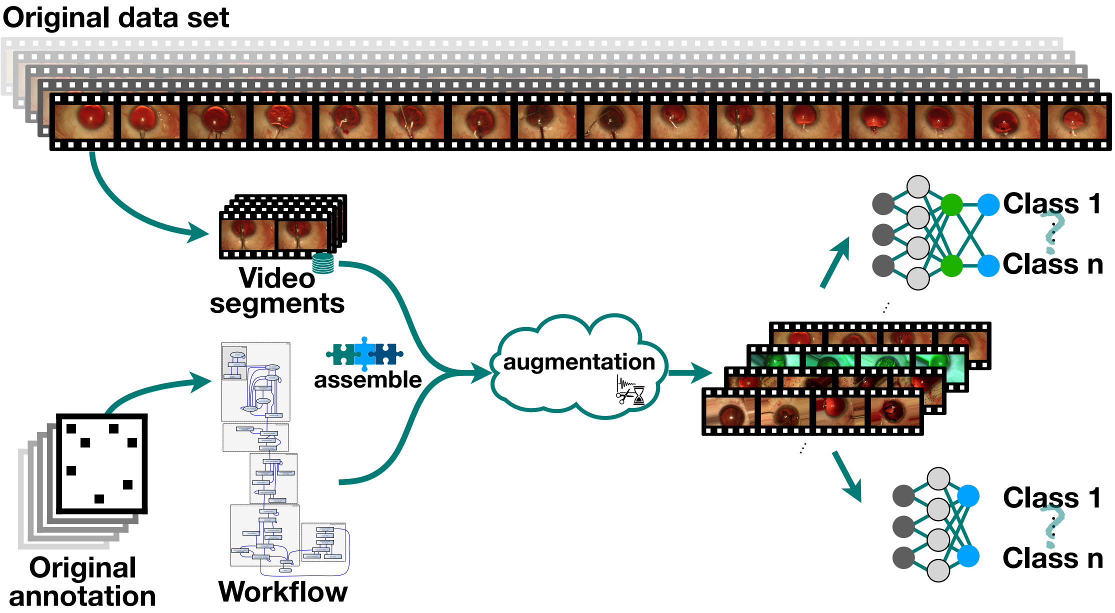


Figure 4.1: Overview of our approach to augmented videos. We extract the workflow from the annotation of the original training dataset and split the videos along their classifications. Afterwards, we assemble new artificial videos using the workflow graph and the sequences. Additionally, they are spatially and timely augmented.

in both space and time, as shown in Figure 4.1. This methodology allows the creation of new artificial videos that appear to the NN as originally recorded videos. Furthermore, it is possible to create sequences of the same duration, with a variation of speed, and a balancing of the classifications within the datasets, a posteriori. To the best of our knowledge, there is no approach so far that can enlarge video datasets in a way that the spatial and the temporal information is augmented, and furthermore balance the dataset. We will demonstrate that with our method, it is possible to augment and balance a cataract video dataset. Afterwards, we train a combined CNN and LSTM network whose image classification performance is better than that of the same NN trained on a dataset augmented using the state-of-the-art method.

This chapter is organized as follows: Section 4.2 provides an overview of the cataract dataset and the current review of the state-of-the-art augmentation methods. We present our methods and the state-of-the-art method we used for augmenting in Section 4.3. Additionally, we describe in this section the networks and training strategies that were used. The results are presented in Section 4.4. The chapter ends with a discussion and conclusions in Section 4.5 and Section 4.6, respectively.

4.2 Related Work

In the following, we describe the current state-of-the-art for data augmentation in general, and especially for time-series, and medical data. Furthermore, we depict the state-of-the-art for classifying time-series data in medicine. We also highlight the differences in previous work compared to our study. Lecun et al. [68] were the first who used data augmentation, i.e., data wrapping, for handwriting recognition using a CNN. Nowadays, data augmentation is a well-established technique for image recognition. These augmentations are based on random spatial image manipulations, such as geometric, color, or noise transformations. Many CNN architectures used different types of these transformations. For example, AlexNet by Krizhevsky [69] used clipping, mirroring, and color augmentation. Hereby, AlexNet produced excellent classification results on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset [47]. Other transformations like scaling and cropping were used in the Visual Geometry Group (VGG) by Simonyan and Zisserman [70] or scaling, cropping, and color augmentation in the Residual Networks (ResNet) by He et al. [71], or translation and mirroring in the DenseNet by Huang et al.[72], or cropping and mirroring in the Inception network by Szegedy et al. [73]. These transformations are applied separately to the individual images. For these applications, the temporal information was not relevant.

In contrast, only a few approaches are published to augment videos. Hajj et al. [15] and Parmar et al. [74] suggest a sub-sampling of the original sequence for video augmentation. The sub-sampling increases the size of the dataset, but the sequences of the videos are identical. Kim et al. [75] suggest a temporal variation to skip some video frames. This results in shorter videos, and therefore to an unwanted NN bias. Ji et al. [76] suggest a time-warping approach, to avoid this. In this case, the data are scaled in the time domain. Nevertheless, there is a major drawback, because Ji et al. applied the method to the extracted feature vector of the input data and not to the original dataset. However, the feature data must be unambiguously human-interpretable for such an approach which is rarely feasible. Therefore, an end-to-end augmentation approach that augments both the spatial and temporal information in the videos is desirable.

4.3 Methods

Our approach is based on the workflow extracted from meta information in the videos. This meta-information, i.e., workflow, is present in the data but not limited to them. We can create a balanced dataset using this meta-information and the extracted video segments. A balanced dataset is one of the most significant advantages of our approach. We apply spatial image augmentation on the complete videos to avoid the fact that the workflow augmentation is not a duplication of data sequences. The spatial augmentation is an effective method to enhance the size and quality of the training data. Image augmentation is crucial to successfully apply deep learning models when only insufficient data are available. In addition, we apply temporal augmentation by using optical flow, not on the extracted feature like Ji et al. [76], in order to generate sub-frames between two frames as Niklaus et al. [77] suggested. The application of optical flow also allowed us to use time-warping in the image domain, and not in the feature domain. Such an end-to-end augmentation for videos, which is easily interpretable by humans, has never been proposed before.

4.3.1 Database

This study aims to demonstrate the potential of our workflow augmentation method for enlarging and balancing an existing dataset. Therefore, we chose the cataract dataset [22] because it has a few videos and is highly imbalanced. Furthermore, the possible tool set is limited due to the high standardization of the cataract surgery, which reduces the dimensions of the workflow graph. We believe that the selection of the cataract dataset as an example does not limit the power and generality of our method. A schematic overview pipeline of our approach is shown in Figure 4.2.

The dataset contains videos from 50 cataract surgeries conducted at the university hospital of Brest. A microscopic video and a surgical tray video were provided for the complete surgical procedure for each surgery. The surgical video was recorded with a ZEISS OPMI Lumera T microscope (Carl Zeiss Meditec AG, Jena, Germany). The videos are stored as a sequence of images. The average duration of the videos is 10 minutes and 56 seconds. The image format is 1920×1080 pixels, and the frame rate was approximately 30 frames per second (fps). The videos of the surgical tray are provided in parallel, as a side-by-side synchronous recording of the surgical tray to the microscope videos. Since the

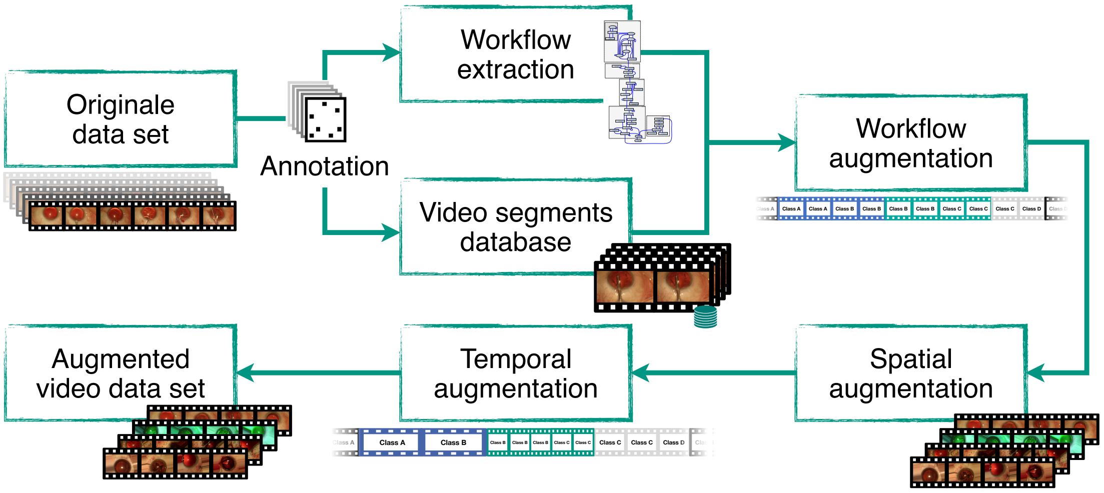


Figure 4.2: Pipeline of the workflow-based augmentation method.

tray videos are not used for this investigation, a more detailed description is omitted and can be found in [22]. The authors also provide the frame-by-frame annotation separately for each video for 21 different surgical tools used or not used. Two experts independently did the annotation regarding the usage of a tool. Here, usage is defined by whether or not the tool is in contact with the eyeball in the respective frame. The annotation of both surgeons was corrected regarding the used tool but not to the exact time of use. Here, the mean of the two experts' annotations is used as the final frame annotation. Therefore, a particular uncertainty in the annotation of the exact point in time must be considered. The full dataset was divided into a training set and a test set of 25 videos each. The training dataset is the starting point for all further steps of our approach.

4.3.2 Augmentation Method

In the following, we describe our end-to-end workflow augmentation methodology for generating new artificial videos in chronological order. First, we explain how we extract the basic workflow from the existing dataset. Next, we describe how we create our modular construction kit of video segments and how we generate new videos with this kit using the workflow as a sequence template. Then we describe our exact procedure for varying the generated videos in appearance and temporal dimension. Afterwards, we describe a state-of-the-art split augmentation method that we use for comparison.

4.3.2.1 Workflow Augmentation

Workflow extraction So that the artificial cataract operations videos mimic a real cataract surgery. We require the workflow of cataract surgery as a template. A workflow is a composed and defined sequence of single work steps. These work steps correspond, in our case, to typical phases of a surgical procedure. A phase of a surgical procedure always refers to a specific goal or general activity, e.g., opening the surgical cavity. We define a *phase* as a higher-level surgery action. When we look at the surgical phase at the next level of detail, we see a sequence of basic actions. In our case, these actions correspond to the events to be classified, namely the used surgical tools. A change in the tool annotation defines the beginning and the end of an event. In the following, the identified events are also named classes. The title of the class indicates the respective tools.

In the literature, various studies on workflow recognition of cataract surgery can be found. However, these workflows varied substantially from each other. For example, Morita et al. [67] propose only a three-phase workflow that contains only the essential steps. Yu et al. [78] propose eight phases, including two additional case differentiation. Zisimopoulos et al. [79] propose 14 phases. The study based on the same dataset as ours. However, Al Hajj et al. [22] suggested 18 phases for the same dataset. Nevertheless, since the phase annotations are not available, we decided to extract the workflow semi-automatically from the annotation files of the training dataset, inspired by [67, 78, 79].Our extracted workflow has six unique phases. These phases are *marking*, *cutting*, *capsulorhexis*, *phacoemulsification*, *implantation*, and *suturing.* The phases are linked by specific transitions, i.e., a specific sequence of events and characterized by different anatomical situations and environmental conditions. Even if identical tools are used in different phases, this leads to a different context in which the tools are used. Consequently, even if they are identical but used in a different phase, these tools are later considered as different tools and classes, respectively. By this, an ambiguity of the tool classification of the NN should be prevented. Consequently, first all classes contained in the entire dataset are identified. The identification has been made only based on the annotation. The class sequence length varies in the 25 videos. The mean length of a video is 27 classes with a standard variation of ±6.5. The maximum length is 130 (video *19*), and the minimum is 18 (video *23*). To extract the workflow the individual sequences of classes were sorted according to their length. Afterwards, they were manually merged to one overall workflow. This step could be done automatically, but further investigation would be necessary. However, this automated workflow extraction goes beyond the scope of this work.
To reduce complexity, we decided to exclude specific surgical phases from the workflow construction, such as IOL removal, which occurs only in the video *12* and *14*. Furthermore, for the stitching with the suture need, only one variation was considered for the workflow. Video *4* was excluded for testing purposes. In the end, we completely excluded the following videos: *4*, *8*, *12*, *14*, *19*, and *25*, due to appearing surgical problems. We know that phases and the tools from the excluded files may not be classified correctly during testing. However, we assume that these simplifications will not limit the methodology. A more complex workflow model would only increase the size of the training dataset and thus the training time.

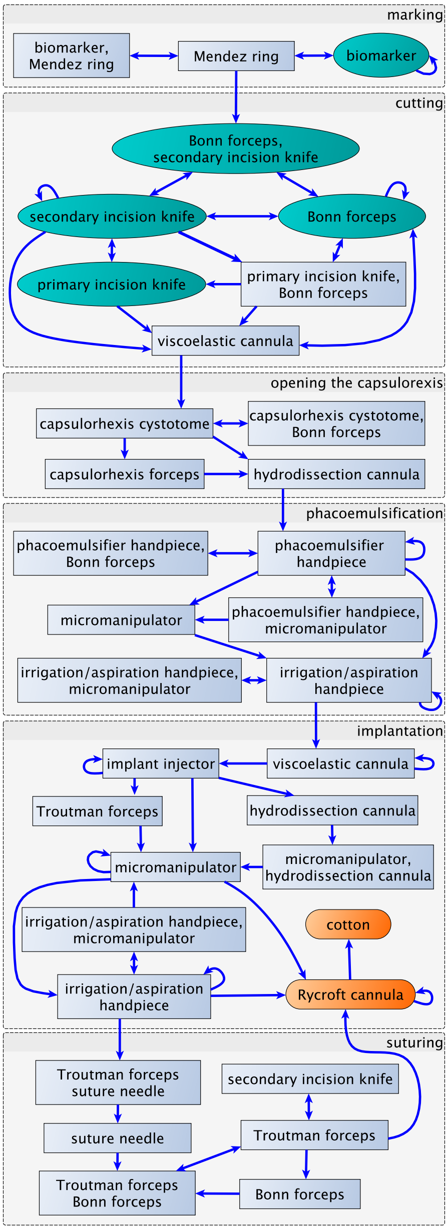

marking
biomarker, Mendez ring
Mendez ring
biomarker
cutting
Bonn forceps, secondary incision knife
secondary incision knife
Bonn forceps
primary incision knife
primary incision knife, Bonn forceps
viscoelastic cannula
opening the capsulorexis
capsulorhexis cystotome
capsulorhexis cystotome, Bonn forceps
capsulorhexis forceps
hydrodissection cannula
phacoemulsification
phacoemulsifier handpiece, Bonn forceps
phacoemulsifier handpiece
micromanipulator
phacoemulsifier handpiece, micromanipulator
irrigation/aspiration handpiece, micromanipulator
irrigation/aspiration handpiece
implantation
implant injector
viscoelastic cannula
Troutman forceps
hydrodissection cannula
micromanipulator
micromanipulator, hydrodissection cannula
irrigation/aspiration handpiece, micromanipulator
cotton
irrigation/aspiration handpiece
Rycroft cannula
suturing
Troutman forceps suture needle
secondary incision knife
suture needle
Troutman forceps
Troutman forceps Bonn forceps
Bonn forceps

Figure 4.3: Illustration of the extracted workflow from the annotation of the training data of the cataract dataset. In green marked the starting classes, in orange the final classes, and square boxes the other detected classes. The gray boxes show the six different phases (marking, cutting, capsulorhexis, phacoemulsification, implantation, and suturing) that we identified from the dataset. The arrows represent the transitions between the classes. For a better overview, the idle intervals are not shown explicitly.

The graph of the final workflow is shown in Figure 4.3. Out of the 25 videos, we identified just four starting classes marked in green in round boxes (*biomarker*, *Bonn forceps*, *primary incision knife*, *secondary incision knife*) and two final classes (*cotton*, *Rycorft cannula*) marked in orange round boxes. The squared boxes are intermediate classes. The arrows represent the transitions between the classes.

Video segment database The workflow and video segments are required to assemble the artificial cataract surgery video. Therefore, the dataset videos have to be split into segments and sorted according to their classification. Before splitting the videos into segments, we had to decide whether the segments should start precisely with the annotation or a few frames before. The best case would be if the segment of a class starts and ends with some frames of the class *no tool in contact*. Unfortunately, this is not possible in the cataract dataset because, for example, if a surgical tool is annotated in the current segment and another one is added, then the class changes without having the classification *no tool in contact* in between. Therefore, we decided to split the video in the middle of the previous and the current class, as illustrated in Figure 4.4. The split at this position has the advantage that a high entropy of a sequence cut, due to the image change, does not bias the NN. More precisely, the NN should not learn to recognize any cuts in the generated video. Instead, the NN should interpret the cuts as noise and does not react sensitively to them because the classification does not change. In addition, it provides more opportunities for a variety of segment combinations in artificial videos. The segments are stored in a database with their frames-by-frames annotation. The current phase is also saved, allowing instruments used multiple times in the workflow to be assembled according to the specific content. We included all videos from the original dataset to have a higher diversity in the database, except video *4*. This video was excluded to have an original clinical test video with annotation because, at that time, the annotations of the 25 test data from the cataract dataset were not publicly available.

As a result, we obtain 124 different types of class transition, for the segment database. The transitions with initial class *no tool in contact* to a class with one or more tools in contact is the largest subset with 45 segments, and vice versa is the second-largest subset with 35 segments. 44 subsets contain just one individual segment. The transition *Rycroft cannula implantation → no tool in contact* contains the largest subset with 127 different segments. In the mean, a class contains approx. 8.78 individual segments. The number of frames per segment is in median 184.5 frames with the median absolute deviation of 253.2 frames. The minimal number of frames per segment is 1, and the maximal number is 3733 frames.

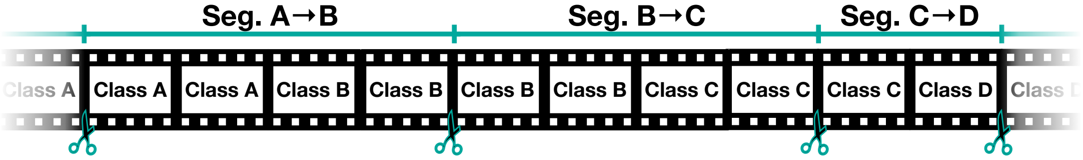


Figure 4.4: Illustration of splitting the original videos in different segment along the classes.

Workflow augmentation The artificial cataract surgical videos were created in the next step based on the workflow graph and the video segment database. The workflow augmentation allows generating videos that have not existed before in the concrete sequence and class combination. The initial and final classes are equivalent to those of the original videos. Furthermore, the transition probabilities between the classes for the artificial videos have to be determined. They can be chosen according to the original dataset, which leads in the case of the Cataract dataset to an unbalanced enlargement or arbitrarily and uniformly distributed for a balanced enlargement. We decided to choose transition probabilities uniformly because we wanted a balanced dataset. Therefore, we chose the outgoing transition probabilities of each class w_i that they sum up to one:

$$1 = \sum_{i=0}^{E} w_i \tag{4.1}$$

To prevent "endless" cycles of one class in the generated artificial videos, the corresponding transition probability was reduced after selection, and the other outgoing transition probabilities were increased simultaneously. The amendment can be determined by the user. We decided to reduce the probability of the selected transition w_j by 1/2. The amendment is then:

$$1 = \frac{w_j}{2} \sum_{i=0, i \neq j}^{E} \frac{w_j}{2 \cdot (N-1)} + w_i \tag{4.2}$$

The transitions of the workflow graph were selected according to their probabilities to generate a new video sequence. This class sequence and the segment database are used to create the corresponding video in the next step. If there is more than one segment for a class, the concrete segment in the database would be randomly but uniformly selected. Since only tool- tool transitions are considered in the workflow graph, but the database also contains sequences with the intermediate step *no tool in contact*, we check in advance which variants are available and then choose randomly between both. As a result, we get a balanced training dataset of artificial videos from an unbalanced original dataset.

Spatial augmentation Since the appearance of the produced videos can be very similar, especially if the database contains only a few sequences, we additionally decided to spatial augment the videos. Therefore, we used 15 different image augmentation types: center-cropped, padding, rotating by 90°, mirroring, zooming, rotating by ±90°, contrast changing, brightness additive or multiplicative changing, gamma-spreading, linearly down-sampling, adding Rician noise, adding Gaussian noise, adding Gaussian blur, and adding square-noise of the batch generator from Isensee et al. [80]. In addition, we used two different types of color augmenting: inverting the colors and a randomly shuffling the color channels. We expect that the color information in the data, e.g., iris color, will be less important. Only geometric shapes, e.g., the instruments or the pupil, strongly influence the correct classification.

Each of the 17 functions is chosen randomly with distribution of 33 %. Afterwards, the execution order is randomly shuffled. This procedure results in a larger variation of the augmentation. This fact can be explained by the circumstance that identical functions with identical parameters, but executed in a different order, led to different results for some functions due to the loss of information. The parameters for the specific augmentation function can be taken from Table 4.1. The values were determined empirically for the dataset, allowing that the user still recognizable content of classification.

Table 4.1: Specific augmentation parameter range of the different functions

function	min	max
center cropping	(840,1080)	(1080,1920)
padding	(1080,2160)	(1920,3840)
90 degree rotation	1	3
mirror axis	x,y	
zoom factor	0.03	1
random rotation	-90	90
contrast	0.2	2
additive brightness	-64	64
multiplicative brightness	0.5	1.5
gamma spreading	0.2	2
linear down-sampling	0.05	2
Rician noise	0	20
Gaussian noise	0	20
Gaussian blur	0	7
square noise	(0,32)	(0,300)

Temporal augmentation After augmenting the appearance of the videos using image augmentation methods, the videos look different, but the video segments have the same duration as the original segments. Since we also consider the temporal information during training, there is still a risk that the NN can memorize the few original video segments, which leads to overfitting. For this reason, we decided to augment the videos also in the temporal domain.

A possible augmentation method would be to increase or decrease the segments' duration by changing the frame rate. A frame rate increase or decrease can be realized by dropping or duplicating individual frames. However, these techniques have a substantiable disadvantage. Dropping or duplicating frames leads to discontinuity of the optical flow in the video. The discontinuity can be recognized by non-natural jumps or non-smooth movements of the surgical tools. As a result, this could negatively affect the network's performance because the videos would have an unnatural character. Moreover, in case of duplication, the inter-frame differences are zero and thus there is no new or different information. For this reason, we decided to use the optical flow to generate new sub-frames. For generating an inter sub-frame from two frames. Therefore, we used the implementation from Niklaus et al. [77]. Niklaus et al. proposed an encoder-decoder network that extracts features from two given frames. These features are the inputs of four NNs. Each sub-network estimates, in a pixel-wise manner, a one-dimensional kernel for each output pixel. Afterwards, the estimated kernels are convoluted with the two input frames to obtain the interpolated sub-frame.

For speed variation, we chose a temporal augmentation range between $0.5\times-1\times-2\times$. The range are divided equally into 20 different factors. These factors are the following: 2, 1.8824, 1.778, 1.6842, 1.6, 1.4884, 1.3913, 1.3061, 1.2075, 1.1637, 1, 0.9552, 0.9014, 0.8533, 0.8, 0.7529, 0.7033, 0.6534, 0.5981, 0.5517 and 0.5. To achieve the different speeds in the videos, the inter-frame space must be up-sampled 64 times. The speed factors then correspond to the following sub-frames: 128, 116, 107, 98, 91, 85, 80, 75, 71, 67, 64, 58, 53, 49, 46, 43, 40, 38, 36, 34, and 32. The full up-sampling can also be applied directly to the segment dataset. This would result in a speedup, especially in the case of a small database or if many artificial videos have to be augmented. For the annotation of the sub-frames $[1, 32]$, the annotation of the original frame n was taken, and for the sub-frames $[33, 64]$, the annotation of the frame $n + 1$. In contrast to the spatial augmentation, we did not augment the entire generated video with one parameter set. Instead, we divided the video into randomly long parts according to the mean plus/minus the mean absolute deviation of the original class sequences. Then, we augmented the individual parts of the video with a randomly selected discrete speed factor. We expect this will lead to greater variability in the duration of individual surgical processes and further to more classification robustness of the NN. To ensure that the complete range for the augmentation length for a video and speed variations is covered as uniformly as possible, we created a Halton sequence. We chose the Halton sequence, more precisely a two-dimensional Halton sequence from $[0-1]$, because it has a small discrepancy, i.e., the sequence is randomly and covers the complete range uniformly. Afterwards, we linearly interpolated the samples to that the first dimension represents the sequence length and the second dimension the speed factor. The concrete augmentation values are randomly selected from the Halton sequence. The temporal augmentation was the last step of our workflow augmentation approach, and as a result we got artificial videos.

4.3.2.2 Comparison Approach: Split Augmentation

Shen et al. [81] and Al Hajj et al. [15] introduced a more simple approach to increase the samples of sufficient datasets artificially. Thereby, they split the videos into smaller subsets. They take just every 10$^{\text{th}}$ frame of the video to get ten videos out of one video. We decided to use this approach as a comparison method. This approach allows easily to artificially increase the amount of data without including other videos or manipulating the content of the videos. In addition, since the individual sub-videos do not differ much from each other, we decided to spatial augment each sub-video individually that they appear differently. For the spatial augmentation, we chose the same procedure and parameters as described in Section 4.3.2.1, Spatial augmentation. Furthermore, for better comparability, we chose only the videos used to create the workflow graph, as described in Section 4.3.2.1, Workflow extraction. These were, in total, 19 original cataract videos, which resulted in 190 sub-videos.

4.3.3 Training strategy

To determine the performance of workflow augmentation versus split augmentation in the study, and to ensure that the differences are only due to augmentation, we used the identical NNs for the different, separate trainings. Furthermore, we first train a CNN and then a combined CNN and LSTM classifier to test whether the additional semantic augmentation provides any benefit. Besides, we converted the floating-point number annotation for each image from or both datasets, the workflow augmented and the split augmented, to a binary annotation. Therefore, labels greater than 0.5 were set to 1, others to 0. Additionally, we transfer the multi-label annotation into a multiclass annotation. This conversion resulted in 27 different classes instead of 21 classes. Furthermore, we normalized the images over the complete dataset. For the training, validation and testing, we divided the 5000 videos of the workflow augmented dataset into 3000 videos (60 %) for training and 2×1000 videos (20 %) for validation and testing, respectively. These test videos were never presented to the classifiers.

For the split dataset, we decided to take all 190 videos for the training and use the 1000 validation videos from the workflow augmented dataset for validation due to the small number of videos in the split dataset.

For both trainings of the NN, we used the stochastic gradient descent (SGD) optimizer with a learning rate of $lr = 0.01$ and a $momentum = 0.9$. The learning rate was linearly decreased by 0.1 every 10th iteration. We trained our models in parallel on 8 NVIDIA Tesla V100s with 32 GB using module-level data parallelism. Due to memory constraints, the CNN classifier is trained on a maximum of 2400 frames whenever possible the complete video is taken. Otherwise, the frames are randomly selected over the entire video. Finally, the selected frames are shuffled. For the training and validation of CNN and LSTM classifier, at least every 15th frame, which corresponds to a frame rate of 2 fps, and a maximum of 2400 frames per video were equidistantly selected from the datasets. The frame sequence was maintained. The videos of the training set were randomly selected from both datasets. For the evaluation, we used the cross-entropy loss and the one-hot classifier. We trained the models until they converged.

4.3.4 Implementation Details of the Neural Networks

CNN network We selected for CNN classifier the ResNet50 [82] network. Because the ResNet50 showed that it was one of the best performing networks at the cataract challenge [61] in comparison with other NNs. Additionally, we decided to use transfer learning due to the limited available training data. We replaced the last fully connected layer with a type-identical layer of 27 nodes representing the 26 binary multi-classes and the class *no tool in contact*. The weight of this layer was randomly initialized. Since the complete dataset was normalized, we decided to disable the estimation of batch normalization for each batch normalization layer. The variable *track_running_stats* was set to *False* and the variables *running_mean* and *running_var* to *None*. This results in a scaling factor γ of 1, and a shift factor β of 0 for each batch normalization layer. Moreover, the batch normalization then based only on the current batch, which representing one complete cataract surgery video. The normalization y of image x is calculated with the following equation:

$$y(x) = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} \,, \tag{4.3}$$

where E is the expected value, Var the variance and ϵ are by default 10^{-5}. For a better overview, all described parameters with the chosen values are also listed in Table 4.2.

Table 4.2: Selected and changed parameters and their values for the training of the network ResNet50

Parameter	**Value**
dataset split in % for workflow augmentation	60/20/20 (train/ valid/ test)
Optimizer	SGD
Classifier	one-hot classifier
Learning rate	$0.01 - 0.1 \cdot \lfloor \frac{\text{epoch}}{10} \rfloor$
Momentum	0.9
Max batch size	2400 frames
Step width for LSTM	15 frames
Batch normalization layer:	
track_running_stats	*False*
running_mean	*None*
running_var	*None*

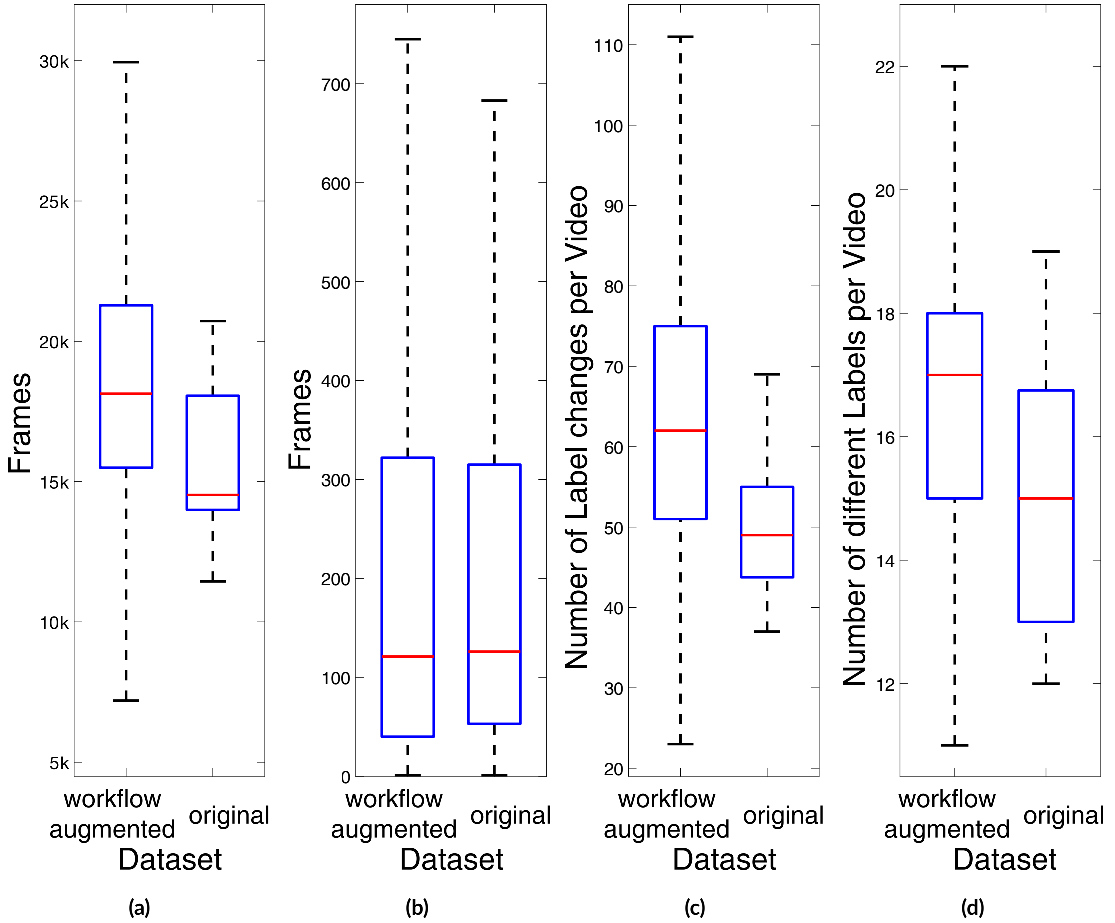
LSTM network To keep the influence of the network selection as low as possible, we extended the previously explained ResNet50 by a layer of 21 LSTM nodes [33] and a fully connected linear output layer with 27 nodes. To avoid learning the network from scratch and make the LSTM layer's training more stable, we used the previously described ResNet50 pre-trained on the images over 8 epochs.

4.4 Results

We conducted three experiments to evaluate the performance of our workflow augmentation approach. The first experiment is the classification performance of the CNN and the combined CNN and LSTM network, in the following briefly called just LSTM, on the workflow augmented test dataset. For the second, we evaluate both NNs on a real test dataset. Third, we evaluate the split trained combined NNs on the same real test dataset. Before we evaluated the experiments, we further investigated the generated dataset produced by our new approach.

4.4.1 Generated Videos of Workflow Augmented Dataset

The generated workflow dataset contains 5000 artificial videos. They are generated with the workflow graph by using 19 original videos and the corresponding segments from 24 videos.



(a) (b) (c) (d)

Figure 4.5: Boxplots of indicators of the workflow augmented dataset, on the left side of each sub-figure, and the original dataset, on the right. Each boxplot shows in red the median and in blue the inter quantile distance 25th percentile and the 75th percentile. In the black dashed line are the whiskers with the 25th percentile minus, resp. 75th percentile plus 1.5 times the inter quantile distance. (a) Shows the plots for the total video length. (b) Shows the plots for the individual sequence length of the videos. (c) Shows the plots for the number of label changes of each video. (d) Shows the plots of the number of different labels for each video.

Table 4.3: Comparison of the dataset augmented by the workflow and original dataset

		25th perc.	median	75th perc.
Total video length (frames)	original	13993	14525	18061
	workflow aug.	15496.5	18134	21282.5
Sequence length (frames)	original	53	126	315
	workflow aug.	40	121	322
# label changes per video	original	43.75	49	55
	workflow aug.	51	62	75
# diff. labels per video	original	13	15	16.75
	workflow aug.	15	17	18

(#): number of

The original videos have a frame count of 14525 in the median with the 25th percentile of 13993 and the 75th percentile of 18061 frames. The videos of the workflow augmented dataset had a length of 18134 frames in the median, with the 25th percentile of 15496.5 and the 75th percentile of 21282.5 frames (shown in Figure 4.5 (a) and Table 4.3). Figure 4.5 (b) and Table 4.3 show a similar distribution for the workflow augmented and the original dataset's segment lengths. The segments in the original videos had the median frame count of 126 with the 25th percentiles of 53 and the 75th percentiles of 315. The segments of the workflow augmented dataset videos had the median length of only 121 frames with the 25th percentile of 40 and the 75th percentile of 322 frames. As a result, we could note, that the original videos are shorter, and the length variability is not uniformly distributed in the workflow augmented dataset. Furthermore, the segment length variability of the original and workflow augmented datasets is very similar.

Furthermore, the workflow augmented dataset had a higher alternation of classifications, with the median of 62 changes and the 25th percentile of 51 and the 75th percentile 75 changes, within a video. The videos in the original dataset had in the median 49 class changes per video and the 25th percentile of 43.75 changes and the 75th percentile of 55 changes (shown in Figure 4.5 (c) and Table 4.3). Also, the number of used tools and tool combinations was greater than in the original dataset, with the median of 17 classes and the 25th percentile of 15 and the 75th percentile of 18. In the original dataset, the median of the classes is 15 and the 25th percentile 13 and the 75th percentile 16.75 classes (shown in Figure 4.5 (d) and Table 4.3). Table 4.4 shows the class distribution in detail of both datasets. The table also indicates that the classes of the workflow augmented dataset are more balanced than those of the original dataset. Most of the classes appeared more frequently in the workflow augmented dataset except the classes: *Rycroft cannula*, *phacoemulsifier handpiece*, *primary incision knife*, *phacoemulsifier handpiece & micromanipulator*, and *irrigation/aspiration handpiece & micromanipulator*. Further, the workflow augmentation approach increased the rare class, e.g., *hydrodissection cannula & micromanipulator*, approximately by a factor of six from 0.039 % of all frames in the original dataset to 0.243 % in the augmented dataset.

Table 4.4: Class distribution of the different datasets in percent

Classes	workf. augment.	org. train	test set
no tool in contact	50.069	46.304	42.687
biomarker	0.084	0.026	0.045
hydrodissection cannula	2.057	2.021	1.843
Rycroft cannula	0.734	3.201	3.382
viscoelastic cannula	2.766	1.614	1.549
cotton	1.417	0.147	0.010
capsulorhexis cystotome	4.930	4.687	5.877
Bonn forceps	0.209	0.184	0.028
capsulorhexis forceps	1.255	0.970	0.537
Troutman forceps	0.758	0.086	0.007
irrigation/aspiration handpiece	17.197	16.769	16.479
phacoemulsifier handpiece	2.410	3.274	3.999
implant injector	1.705	1.464	1.522
primary incision knife	0.429	0.448	0.576
secondary incision knife	0.464	0.296	0.373
micromanipulator	2.176	1.596	1.483
suture needle	0.110	0.027	0.000
Mendez ring	0.448	0.162	0.133
Mendez ring & biomarker	0.003	0.001	0.014
Bonn forceps & secondary incision knife	0.320	0.292	0.255
primary incision knife & Bonn forceps	0.567	0.324	0.174
capsulorhexis cystotome & Bonn forceps	2.781	0.361	0.028
phacoemulsifier handpiece & Bonn forceps	0.083	0.052	0.129
phacoemulsifier handpiece & micromanipulator	5.426	13.336	14.441
irrigation/aspiration handpiece & micromanipulator	1.229	2.280	4.429
hydrodissection cannula & micromanipulator	0.243	0.039	0.000
Troutman forceps & suture needle	0.127	0.040	0.000

4.4.2 Classification of the Neural Networks on the Workflow Augmented Test Dataset

The first experiment is designed to test performance of both classifiers. We trained the CNN classifier without the temporal information on the workflow augmented dataset to have a baseline performance for comparison. Therefore, the classifiers were trained on workflow augmented data until converges. That results in a validation loss: 0.0759 and an accuracy (ACC): 97.78 %, after 42 epochs of max. 50 epochs for the CNN and 38 epochs of 50 epochs with an evaluation loss: 0.952 and ACC: 97.49 %, for the LSTM.

Eight epochs take for the ResNet50 approx. one day and one epoch per day for the LSTM on 8 NVIDIA Tesla V100 GPUs with 32GB RAM used in parallel.

We evaluated both classifiers on the workflow augmented test dataset. Therefore, we choose every 15$^{\text{th}}$ image of the video due to above mention memory issues. For each NN, we evaluated the following overall metrics for multi-class imbalanced datasets [83] for each NN: ACC, mean accuracy (AvACC), class balanced accuracy (CBA), macro-mean recall (REC$_M$), macro-mean precision (PREC$_M$), and macro F1-score (F1-score$_M$). For the calculation, the classes which do not occur in the dataset are excluded.

CNN The CNN classifier predicted the classes with an overall ACC of 96.9 % and an AvACC of 99.77 %. The CBA was lower, with 87.41 %. With a PREC$_M$ of 96 %, the classifier hits the classification correctly. The REC$_M$ was 87.5 %, and the F1-score$_M$ was 95.56 %.

Table 4.5 shows the ACC, precision (PREC), recall (REC), specificity (SPEC), and F1-score for the binary analysis of the classification. The respective class was evaluated against all others aggregated into one class. In other words, the table shows the classifier's ability to recognize the presence or absence of a specific tool class. The classifier predicted the classes correct with a probability over 99.2 %, except the class *no tool in contact* with an ACC of 97.85 %. The PREC of all classifications was over 90 %, and the RECs were over 74 %, with two exceptions: *biomarker* with 67.9 % and *Mendez ring & biomarker* with 19.5 %. In contrast, the SPECs were above 99 % with one exception: the class *no tool in contact* with 96.7 %. The table also shows that the F1-scores were above 77.76 % except for the *Mendez ring & biomarker* of 32.65 %. Additionally, we provide the complete confusion matrix for the CNN classifier on the workflow augmented test data in Table A.1.

LSTM The LSTM classifier showed in general similar results as the CNN classifier. The LSTM classifier predicted the classes correctly with an ACC of 96.6 % and a AvACC of 99.75 %. All values, the CBA with 88.47 %, the PREC$_M$ with 95.86 %, and the F1-score$_M$ with 92.08 % were lower, compared to the CNN classifier. Just, the REC$_M$ with 88.59 % was a higher for the LSTM classifier.

Further, we also did a binary evaluation for the LSTM classifier, shown in Table 4.5. Here the ACC scores were in the same range, from 97.5 % (*no tool in contact*) to 100 % (*Mendez ring & biomarker*), as for the CNN classifier. PRECs and RECs were above 86.9 % and 74.8 %, respectively, which are lower than for CNN, excluding the outlier. However, the REC outlier for the LSTM classifier was lower at 49.88 % (*biomarker*). The SPECs were above 99.5 % for the LSTM with the same exception of the class *no tool in contact* with 96 % as for CNN. The table also shows that the F1-scores were above 81.46 %, except for the *biomarker* with 65.14 %. We also provide the complete confusion matrix for the LSTM classifier in Table A.2.

Table 4.5: Binary analysis of the classification for the CNN and LSTM classifier on the test data of the workflow augmented dataset with the scores for accuracy (ACC), precision (PREC), recall (REC), specificity (SPEC), and F1-score, rounded on 4 digits.

	CNN					LSTM				
	ACC	PREC	REC	SPEC	F1-score	ACC	PREC	REC	SPEC	F1-score
no tool in contact	0.9785	0.9668	0.9907	0.9666	0.9786	0.9750	0.9601	0.9908	0.9595	0.9752
biomarker	0.9997	0.9080	0.6799	0.9999	0.7776	0.9995	0.9385	0.4988	1	0.6514
hydrodissection cannula	0.9976	0.9487	0.9307	0.9990	0.9396	0.9974	0.9444	0.9239	0.9989	0.9340
Rycroft cannula	0.9980	0.9417	0.7879	0.9996	0.8579	0.9977	0.9341	0.7488	0.9996	0.8313
viscoelastic cannula	0.9966	0.9461	0.9307	0.9985	0.9383	0.9961	0.9385	0.9200	0.9983	0.9291
cotton	0.9981	0.9685	0.9001	0.9996	0.9331	0.9963	0.9820	0.7581	0.9998	0.8556
capsulorhexis cystotome	0.9963	0.9785	0.9463	0.9989	0.9621	0.9967	0.9682	0.9637	0.9984	0.9659
Bonn forceps	0.9993	0.9143	0.7487	0.9999	0.8232	0.9993	0.8692	0.7665	0.9998	0.8146
capsulorhexis forceps	0.9989	0.9770	0.9408	0.9997	0.9585	0.9993	0.9836	0.9659	0.9998	0.9747
Troutman forceps	0.9990	0.9657	0.9039	0.9997	0.9337	0.9989	0.9671	0.8912	0.9998	0.9276
irrigation/aspiration handpiece	0.9920	0.9776	0.9762	0.9953	0.9769	0.9921	0.9797	0.9745	0.9958	0.9771
phacoemulsifier handpiece	0.9975	0.9488	0.9458	0.9988	0.9473	0.9972	0.9652	0.9174	0.9992	0.9407
implant injector	0.9985	0.9707	0.9400	0.9995	0.9551	0.9986	0.9736	0.9435	0.9996	0.9583
primary incision knife	0.9991	0.9580	0.8494	0.9998	0.9004	0.9990	0.9317	0.8342	0.9997	0.8803
secondary incision knife	0.9989	0.9221	0.8184	0.9997	0.8671	0.9988	0.9078	0.8116	0.9996	0.8570
micromanipulator	0.9971	0.9506	0.9174	0.9989	0.9337	0.9973	0.9439	0.9304	0.9988	0.9371
suture needle	0.9997	0.9479	0.7575	1	0.8421	0.9997	0.9536	0.7258	1	0.8243
Mendez ring	0.9996	0.9810	0.9293	0.9999	0.9545	0.9996	0.9823	0.9375	0.9999	0.9594
Mendez ring & biomarker	1	1	0.1951	1	0.3265	1	1	0.9756	1	0.9877
Bonn forceps & secondary incision knife	0.9994	0.9177	0.9020	0.9997	0.9098	0.9994	0.9395	0.8707	0.9998	0.9038
primary incision knife & Bonn forceps	0.9994	0.9629	0.9356	0.9998	0.9491	0.9993	0.9675	0.9121	0.9998	0.9390
capsulorhexis cystotome & Bonn forceps	0.9994	0.9966	0.9818	0.9999	0.9892	0.9996	0.9984	0.9864	1	0.9924
phacoemulsifier handpiece & Bonn forceps	0.9999	0.9596	0.9344	1	0.9468	0.9999	0.9393	0.9134	1	0.9261
phacoemulsifier handpiece & micromanipulator	0.9978	0.9881	0.9717	0.9993	0.9798	0.9974	0.9944	0.9592	0.9997	0.9765
irrigation/aspiration handpiece & micromanipulator	0.9984	0.9798	0.8976	0.9997	0.9369	0.9984	0.9739	0.9070	0.9997	0.9393
hydrodissection cannula & micromanipulator	0.9999	0.9887	0.9836	1	0.9861	0.9999	0.9882	0.9845	1	0.9864
Troutman forceps & suture needle	0.9998	0.9549	0.9287	0.9999	0.9416	0.9998	0.9586	0.9068	0.9999	0.9320

We can conclude a first result, that the accuracies of the two NNs are similar and do not differ meaningfully from each other. Furthermore, we can note that the complexity of the CNN is high enough to be able to extract appropriate features and thereby correctly classify the tools and tool combinations. However, from these similar results of the NNs, we cannot deduce if the temporal component, that we aimed to augment with our approach, has any influence or was considered in the classification for the LSTM classifier. The good results for both classifications could be due to the small differences within the complete workflow augmented dataset. Therefore, we test both classifiers again using original clinical videos. For this purpose, we take the excluded video of the original training dataset and the original test dataset from the Cataract dataset. [22] to retest the NNs.

4.4.3 Classification of the Neural Networks on real Surgical Videos

Al Hajj et al. [22] provided during our study the annotation of the test dataset from the Cataract dataset. Therefore, we had the choice to test our classifiers on further real cataract surgery videos and not only on the video *4* which we excluded for test purposes from the augmentation procedure.

To perform a representative test, we must determine which tools and tool combinations are present in the test data because not all were present in our training dataset. The selection is important because classes that were not included in the training data can strongly negatively bias the results. Especially if the temporal sequence is considered and the spatial information in the individual frames, as for the LSTM network. After evaluating the containing tool classes, we had to excluded eight of the 25 videos. The real test dataset now contains only 17 in addition to video *4* of the training dataset. Table 4.4 shows for each class the distribution of the selected test videos. However, the test dataset does not include all instrument classes. The following classes were not observed in the selected videos: *suture needle*, *hydrodissection cannula & micromanipulator*, and *Troutman forceps & suture needle.*

4.4.3.1 Trained on the Workflow Augmented Dataset

In the following, we present the results of the two classifiers trained on the workflow augmented dataset and retested on the real video.

CNN On the real test dataset, the CNN classifier predicted the classes with an ACC of 82.12 % and an AvACC of 98.68 %. The CBA was 39.4 %, excluding the classes where row and column are zero, i.e., *suture needle*, *Troutman forceps & suture needle*, which can be identified in Table A.3. The mean PREC_M was 64.29 %. The mean REC_M was 44.89 %, and the F1-score_M was 52.86 %. The CNN classifier results on real videos were markedly worse than for the workflow augmented test dataset. The ACC was 12 percentage points (pp) lower on the real video than on the workflow augmented test videos. All other scores were also lower.

The binary analysis in Table 4.6 shows that the classes were detected with an ACC over 94.14 %. The class *no tool in contact*, which occurred mostly in the training data, had the lowest ACC with 90.23 %. However, the high ACC was not achieved by the correctly recognized classes. Instead, it was achieved by correctly classifying the non-classes, as seen in the PREC and REC scores. The PREC can partly not be calculated, i.e., *biomarker*, *Mendez ring*, or *Mendez ring & biomarker*, because these classes were never predicted. The PREC scores for these classes were *n/a.* For the classes *Bonn forceps*, *Troutman forceps*, *capsulorhexis cystotome & Bonn forceps*, and *hydrodissection cannula & micromanipulator*, the PRECs were zero due to less than 5 false-positive predictions per class. The PRECs ranged from 44.64 % to 100 % for the other classes. The REC was in the range 0 % to 98.54 %. Thereby, 11 classes were 0 % because these classes were never correctly classified, which can be observed in the confusion matrix for the CNN classifier in Table A.3. The CNN classified these classes as the class *no tool in contact.* Furthermore, the SPEC for the class *no tool in contact* with 84.43 %, also the lowest score compared to the other classes. Here, the SPECs were above 94 %.

Table 4.6: Binary analysis of the classification for the workflow trained CNN and LSTM classifier on the original test dataset with the scores for accuracy (ACC), precision (PREC), recall (REC), specificity (SPEC), and F1-score, rounded on 4 digits. Entries with *n/a* could not be calculated, because they were not predicted and therefore, they are 0.

	CNN					LSTM				
	ACC	PREC	REC	SPEC	F1-score	ACC	PREC	REC	SPEC	F1-score
no tool in contact	0.9023	0.8153	0.9854	0.8443	0.8923	0.9673	0.9426	0.9801	0.9584	0.9610
biomarker	0.9995	*n/a*	0	1	0	0.9995	*n/a*	0	1	0
hydrodissection cannula	0.9929	0.8210	0.7992	0.9966	0.8100	0.9973	0.9313	0.9242	0.9987	0.9278
Rycroft cannula	0.9778	0.9375	0.3870	0.9991	0.5478	0.9881	0.9109	0.7276	0.9974	0.8090
viscoelastic cannula	0.9871	0.5642	0.8311	0.9896	0.6721	0.9961	0.8833	0.8694	0.9981	0.8763
cotton	0.9999	*n/a*	0	1	0	0.9999	*n/a*	0	1	0
capsulorhexis cystotome	0.9818	0.9510	0.7375	0.9976	0.8308	0.9949	0.9562	0.9590	0.9972	0.9576
Bonn forceps	0.9997	0	0	1	0	0.9997	0	0	1	0
capsulorhexis forceps	0.9962	0.7526	0.4740	0.9991	0.5817	0.9980	0.9900	0.6429	1	0.7795
Troutman forceps	0.9997	0	0	0.9998	0	0.9999	*n/a*	0	1	0
irrigation/aspiration handpiece	0.9454	0.7827	0.9377	0.9469	0.8533	0.9814	0.9131	0.9837	0.9809	0.9471
phacoemulsifier handpiece	0.9804	0.8818	0.6056	0.9965	0.7181	0.9943	0.9295	0.9319	0.9970	0.9307
implant injector	0.9947	0.8186	0.8486	0.9970	0.8333	0.9978	0.9820	0.8761	0.9997	0.9261
primary incision knife	0.9970	0.8704	0.5697	0.9995	0.6886	0.9971	0.8268	0.6364	0.9992	0.7192
secondary incision knife	0.9971	0.7500	0.3645	0.9995	0.4906	0.9976	0.8421	0.4486	0.9997	0.5854
micromanipulator	0.9887	0.6115	0.7035	0.9931	0.6543	0.9962	0.8997	0.8447	0.9985	0.8714
suture needle	1	*n/a*	*n/a*	1	*n/a*	1	*n/a*	*n/a*	1	*n/a*
Mendez ring	0.9986	*n/a*	0	1	0	0.9987	1	0.0263	1	0.0513
Mendez ring & biomarker	0.9999	*n/a*	0	1	0	0.9999	*n/a*	0	1	0
Bonn forceps & secondary incision knife	0.9988	0.7191	0.8767	0.9991	0.7901	0.9988	0.7941	0.7397	0.9995	0.7660
primary incision knife & Bonn forceps	0.9980	0.4464	0.5000	0.9989	0.4717	0.9985	0.6000	0.4800	0.9994	0.5333
capsulorhexis cystotome & Bonn forceps	0.9996	0	0	0.9999	0	0.9997	0	0	1	0
phacoemulsifier handpiece & Bonn forceps	0.9989	1	0.1622	1	0.2791	0.9987	1	0.0270	1	0.0526
phacoemulsifier handpiece & micromanipulator	0.9414	0.9430	0.6443	0.9932	0.7655	0.9888	0.9512	0.9749	0.9913	0.9629
irrigation/aspiration handpiece & micromanipulator	0.9672	0.8352	0.3475	0.9967	0.4908	0.9818	0.9320	0.6478	0.9977	0.7643
hydrodissection cannula & micromanipulator	1	0	*n/a*	1	0	1	*n/a*	*n/a*	1	*n/a*
Troutman forceps & suture needle	1	*n/a*	*n/a*	1	*n/a*	1	*n/a*	*n/a*	1	*n/a*

LSTM The classification results of the LSTM were also worst on the videos of the real test dataset than on the workflow augmented test dataset. Nevertheless, they were much better than the results of the CNN. The overall ACC was 93.49 %, which was over 11 pp higher than the ACC of the CNN. The AvACC was 99.52 %, and the CBA was 52.43 % excluded classes, for which the rows and columns were zero. The PREC$_M$ was 81.42 %, which was better than the CNN with over 17 pp. The mean REC$_M$ was 53 %, and the F1-score$_M$ was 64.2 %.

In the binary analysis of the classification, shown in Table 4.6, the LSTM generally performed better than the CNN. The ACC was higher than 96.73 % for every class. However, the class *no tool in contact* also had the worst recognition rate. Furthermore, the PREC was only in two classes 0 % and for four classes *n/a* besides those classes which were not included in the test videos. For the other classes, the PREC was at least above 60 % (*primary incision knife & Bonn forceps*), but in most cases above 82 %. The high ACCs of the LSTM classifier was not based on the high true-negative rates like for the CNN classifier, instead to the higher true-positive rates. The scores were either higher or at least equivalent for each classification, except for two classes: *primary incision knife & Bonn forceps* and *phacoemulsifier handpiece & Bonn forceps*, which had lower scores. Table A.4 shows if a frame was misclassified, it was mostly placed in the class *no tool in contact* or a neighboring class, e.g., the class *secondary incision knife* was classified as *Bonn forceps & secondary incision knife* in 12 cases. More examples can be found in the table. In contrast to Table A.3, the misclassification scatters of the LSTM were lower than that of the CNN.

After the second experiment we can conclude that the LSTM must have considered the temporal information during the training. Otherwise, a difference between CNN and LSTM networks would not be observable and explainable. Having said this, another question raised up: can the results also be achieved with a less complicated method, like the split data augmentation? We answer this question with our third experiment.

4.4.3.2 Trained on the Split Dataset

To benchmark the results of our approach, we retrained and evaluated both network CNN and LSTM described in Section 4.3.3. We used the split dataset, augmented based on the video splitting approach as described in Section 4.3.2.2. We applied the same termination criterion for training as for the workflow augmented dataset. The ACC of the CNN converged after 16 of 50 epochs with 98.54 % and a loss of 0.1092 for the training dataset, and an ACC of 55.18 % for the validation dataset. The ACC of the LSTM net converged after 19 of 50 epochs with 98.61 % and a loss of 0.0823 for the training dataset and on the validation data with an ACC of 80.46 %. One epoch takes for the CNN approx. 1 hour and for the LSTM approx. 1.5 hours on 8 NVIDIA Tesla V100 GPUs used in parallel was significantly faster than for other datasets. Afterwards, we evaluated the NNs using the same parameters on the real test data as described previously.

CNN The CNN achieved an overall ACC of 75.52 % and an AvACC of 98.11 %. The CBA was 30.94 %. Here, the ACC and the CBA were lower with about 8 pp as the scores of the CNN trained on the workflow augmented data. The AvACC was also lower but in the same order of magnitude. The scores for PREC_M with 47.72 %, the REC_M with 35.68 %, and the F1-score_M with 40.83 % show that the classifier generally performs worse than the workflow trained CNN.

The binary analysis, shown in Table 4.7, arise that the ACCs for the different classes were in the same order of magnitude as the ACCs of the workflow trained CNN, but the PRECs do not achieve more than 85.42 %. In comparison, the PREC was 0 % for fewer classes, but for example, for the class *phacoemulsifier handpiece & Bonn forceps*, the PREC was 85 pp lower than for the other CNN. The median for the PRECs was 66.49 %. The scores for the RECs were generally lower than for the workflow trained CNN, but there are two exceptions for class *Rycroft cannula* and *irrigation/aspiration handpiece & micromanipulator*. Here, the REC increased to 52.73 % (previously 38.7 %) and 40.19 % (previously 34.75 %), respectively. The scores for PRECs and RECs were lower for the CNN trained on the split augmented dataset than for the CNN trained on the workflow augmented dataset. The values for the F1-score were also lower for the individual classes except for the two classes *Rycroft cannula* and *irrigation/aspiration handpiece & micromanipulator*. The SPECs were like those of the workflow trained CNN. The confusion matrix Table A.5 seen in Appendix A.1 also shows more entries that are not 0 as in Table A.1.

LSTM The LSTM classifier achieved an ACC of 90.87 % on the real surgical videos and an AvACC of 99.32 %. The CBA was 44.24 %. Theses scores are lower than the scores of the LSTM trained on the workflow augmented dataset. Also, for PREC_M with 75.48 %, the REC_M with 45.38 %, and the F1-score with 56.68 % showed that the classifier generally performs worse than the LSTM trained on the workflow augmented videos, which were at least ≈ 6 pp higher. The binary analysis of the split trained LSTM, shown in Table 4.7, yielded similar scores for the ACCs of the classes as the LSTM trained on the workflow augmented videos. The scores of PREC were mostly worse. Only five classes reached scores above 90 %, compared to ten before. In addition, the PRECs could not be calculated for eight classes in addition to the three, which could not be observed in the real test data. They were neither true nor false-positive predicted by the network, seen in Table 4.7. Similar results could also be obtained for the REC scores. Thereby, the scores were 0 % for eight classes, and for three classes above 90 % instead of six. This trend was also visible for the F1-scores.

Furthermore, in the confusion matrix Table A.6 in Appendix A.1, it can be observed that more entries were not 0 as the comparison confusion matrix Table A.2 of the other trained LSTM.

Finally, we conclude that the NNs that were trained on the workflow augmented dataset was the best. In addition, we can also observe that workflow augmentation of temporal

Table 4.7: Binary analysis of the classification for the split trained CNN and LSTM classifier on real surgical videos with the scores for accuracy (ACC), precision (PREC), recall (REC), specificity (SPEC), and F1-score, rounded on 4 digits. Entries with *n/a* could not be calculated, because they were not predicted and therefore, they are 0.

	CNN					LSTM				
	ACC	PREC	REC	SPEC	F1-score	ACC	PREC	REC	SPEC	F1-score
no tool in contact	0.9050	0.8335	0.9607	0.8662	0.8335	0.9619	0.9391	0.9700	0.9562	0.9543
biomarker	0.9995	*n/a*	0	1	*n/a*	0.9995	*n/a*	0	1	0
hydrodissection cannula	0.9791	0.4522	0.4924	0.9885	0.4522	0.9938	0.8447	0.8239	0.9971	0.8341
Rycroft cannula	0.9742	0.6611	0.5273	0.9903	0.6611	0.9808	0.6907	0.8091	0.9870	0.7452
viscoelastic cannula	0.9762	0.3744	0.7387	0.9800	0.3744	0.9915	0.7353	0.7320	0.9957	0.7336
cotton	0.9999	*n/a*	0	1	*n/a*	0.9999	*n/a*	0	1	0
capsulorhexis cystotome	0.9657	0.8542	0.5220	0.9943	0.8542	0.9886	0.9145	0.8955	0.9946	0.9049
Bonn forceps	0.9997	*n/a*	0	1	*n/a*	0.9997	*n/a*	0	1	0
capsulorhexis forceps	0.9941	0.3830	0.1169	0.9990	0.3830	0.9954	0.6582	0.3377	0.9990	0.4464
Troutman forceps	0.9994	0	0	0.9995	0	0.9999	*n/a*	0	1	0
irrigatio*n/a*spiration handpiece	0.9138	0.7222	0.7978	0.9374	0.7222	0.9819	0.9318	0.9636	0.9856	0.9474
phacoemulsifier handpiece	0.9625	0.5526	0.4677	0.9838	0.5526	0.9901	0.9274	0.8246	0.9972	0.8730
implant injector	0.9917	0.7164	0.7821	0.9951	0.7164	0.9959	0.8812	0.8509	0.9982	0.8658
primary incision knife	0.9956	0.7590	0.3818	0.9993	0.7590	0.9952	0.5812	0.6727	0.9971	0.6236
secondary incision knife	0.9969	0.7442	0.2991	0.9996	0.7442	0.9967	0.5660	0.5607	0.9983	0.5634
micromanipulator	0.9811	0.4056	0.5106	0.9884	0.4056	0.9881	0.6186	0.5647	0.9946	0.5904
suture needle	1	*n/a*	*n/a*	1	*n/a*	1	*n/a*	*n/a*	1	*n/a*
Mendez ring	0.9986	0	0	1	0	0.9986	*n/a*	0	1	0
Mendez ring & biomarker	0.9999	*n/a*	0	1	*n/a*	0.9999	*n/a*	0	1	0
Bonn forceps & secondary incision knife	0.9980	0.6545	0.4932	0.9993	0.6545	0.9980	1	0.2192	1	0.3596
primary incision knife & Bonn forceps	0.9977	0.3684	0.4200	0.9987	0.3684	0.9982	0	0	1	0
capsulorhexis cystotome & Bonn forceps	0.9996	0	0	0.9999	0	0.9997	*n/a*	0	1	0
phacoemulsifier handpiece & Bonn forceps	0.9980	0.1538	0.1081	0.9992	0.1538	0.9987	*n/a*	0	1	0
phacoemulsifier handpiece & micromanipulator	0.9009	0.7203	0.5433	0.9632	0.7203	0.9845	0.9515	0.9439	0.9916	0.9477
irrigatio*n/a*spiration handpiece & micromanipulator	0.9636	0.6649	0.4019	0.9903	0.6649	0.9809	0.8367	0.7226	0.9933	0.7755
hydrodissection cannula & micromanipulator	0.9997	0	*n/a*	0.9997	0	1	*n/a*	*n/a*	1	*n/a*
Troutman forceps & suture needle	1	*n/a*	*n/a*	1	*n/a*	1	*n/a*	*n/a*	1	*n/a*

information provides measurable benefits. The artificially generated videos for the NNs are indistinguishable from real surgical videos and even enhance the recognition performance of the classifiers.

4.5 Discussion

One goal of our workflow augmentation approach was to create new artificial videos with a comparable duration to the original videos. This goal could be achieved by choosing the initial transition probabilities uniformly. The chosen probabilities resulted in videos whose duration is more balanced, and the alternation frequency of classes is higher. Additionally, the videos having a very similar length of the individual segment, see Table 4.3. The median segment length is slightly shorter, with 121 frames than 126 frames of the original dataset. The sub-frames variation of temporal augmentation can explain the shorter length. The variation has a median score of 67.5 instead of 64 frames. However, this should not lead to any noticeable effect on the NNs.

A further aim of our approach was that the prevalence of the individual classes within the dataset would be more balanced. So that the recognition performance of classes, which were underrepresented in the original dataset, would be improved. The results in Section 4.4.1 showed that our workflow augmented dataset is more balanced than the original dataset, although not for all classes. The result can also be obtained from Table 4.4. For example, the percentage of the class *Troutman forceps* was increased by a factor of 8.8, but the presence of the previously dominated class *no tool in contact* was also increased from 46.3 % to 50.06 %. However, the relatively underrepresented classes now appear more frequently, but the distribution of the classes is still not uniform. The goal of equal class distribution might be too ambitious and could not be achieved prospectively with the chosen transition probabilities. The underlying goal of better detection of underrepresented classes could perhaps still be achieved by our approach.

Let's look at the different experiments in detail. The first experiment in Section 4.4.2 shows that the selected ResNet50 can separate and classify the individual classes. Unfortunately, the addition of the LSTM layer did not yield any significant improvements for the test videos of the workflow augmented dataset. An explanation for that could be because CNN had already generalized very well due to the many training data images, averaging 43.5 million or the inter-video variability are slightly too low in the workflow augmented dataset. However, this experiment cannot answer whether underrepresented classes are better detected by training the network with a dataset augmented by our approach because the ACCs of the binary analysis for all classes is above 99 %, except for the class no tool in contact. Also, the scores for the PRECs and the RECs are not unambiguous. This also has validity for the LSTM classifier.

However, Section 4.4.3, the second experiment, shows considerable differences from the first experiment's results. From Table 4.6, we conclude that the additional LSTM layer improves the recognition performance. All binary classification ACCs of the LSTM network are at least as high or higher than the one for the CNN. The F1-scores are also higher for the LSTM network, except the class *Bonn forceps & secondary incision knife* and the class *phacoemulsifier handpiece & Bonn forceps*. Here, the F1-cores are lower for the LSTM than for the CNN network. This can be explained by the worse score for the RECs. The class *phacoemulsifier handpiece & Bonn forceps* is also an example of an underrepresented class with 8.3 %. It is the second least common class next to the *Biomarker* class, the third least class, and the *Mendez ring & biomarker* the rarest class. In contrast to the class *phacoemulsifier handpiece & Bonn forceps*, the PREC for *Biomarker* and *Mendez ring & biomarker* could not be calculated because these classes were not predicted by the LSTM network, seen in Table A.4. Furthermore, classes that occur more frequently in the training data but very rarely in the real surgical videos, like *cotton* or *Troutman forceps*, were also not predicted by the network. These classes are mostly detected exclusively as *no tool in contact*, see Table A.4. This misclassification leads to the assumption that the tool's timing in tissue contact is not 100 % correct or that the network has learned to identify every image that is not recognized as a different class as *no tool in contact*. Furthermore, these would be also explained that the class *no tool in contact*, which was most common in the training data, in the test data of the workflow augmented dataset, and in the real surgical videos, performed worst for the ACC in the binary evaluations.

Compare to the study of Al Hajj et al. [15], both NNs trained on the workflow augmented dataset, created by our approach and tested on the real surgical videos, perform better. The mean precision of the CNN network is 64.29 % and 81.42 % for the LSTM network. This is higher than the pure CNN and the combined CNN and RNN from [15], which had a mean precision between 52.93 % and 60.86 % for the CNN and 79.80 % for the combined network. Furthermore, in contrast to Al Hajj et al., we did not perform hyperparameter tuning.
Nevertheless, the results are not directly comparable because not all instruments were included in our workflow graph. For a better comparison, we used the same augmentation method described in [15] with the same conditions as we used to create the workflow augmented dataset, e.g., types of instruments, spatial augmentation parameter, and retrain both NNs on this data.

In that third experiment, we also access a better recognition performance for the trained NNs with the dataset augmented by our approach. The overall ACC for CNN is 82.12 % higher than 75.52 % from the CNN trained on the split dataset. Also, PREC$_M$ with 64.29 % and REC$_M$ with 44.89 % are compared to 47.72 % and 35.68 %, respectively, are higher for the CNN trained on the workflow augmented data. The differences in the scores lead us to assume that the split dataset is, in general, too small for suitable training of a NN like the ResNet50. For the LSTM network, we also obtained remarkably better results for the trained model using our new approach. The ACC is 93.49 % for the NN trained workflow augmented dataset compared to 90.87 % for the split trained. Also, the PrecM with 81.42 %, the RecM with 53 % is higher than 75.48 % and 45.38 % for the state-of-the-art augmentation method.

In contrast to the CNN networks, we could augment the image information and the temporal sequence. Otherwise, the results of CNN and LSTM would not be so different. Finally, we state that the classifier trained on data augmented with our approach provides substantially better results in terms of tool recognition than the state-of-the-art method.

However, we also need to address some limitations. Not all training videos were considered when creating the workflow diagrams. Therefore, direct comparisons cannot be made without constraints on the other publication. Also, no explicit thresholds are used for classification. Instead, the one-hot classifier was used, which means that the threshold for a class can be different for each image, but the score for the predicted class must be the highest. Furthermore, a multi-class approach was used instead of a multiple-label approach, which should not result in any differences. Also, a weighted cost function could result in further improvements, as the dataset is more balanced than the original dataset, but the balancing of the dataset is still not enough.

4.6 Conclusion

This work introduces a novel approach for augmenting videos for video-based event recognition. This methodology allows the creation of new artificial videos that appear to the NN as originally recorded videos. Furthermore, it is possible to create sequences of the same duration, with a variation of speed, and balance the classifications in datasets a posteriori.

Furthermore, the proposed approach has two novelties. The first novelty is that a combined CNN and RNN can be trained end-to-end since sufficient data can now be generated for training. The second novelty is the methodology of augmentation: combining meta knowledge with the workflow, spatial and temporal augmentation makes it possible to balance and completely augment temporal sequences such as videos of a dataset. The proposed methodology is general and applicable outside the scope of cataract surgery.

We have shown that our approach can extend and balance small datasets, using only information contained in the dataset. Furthermore, our method is not limited to only this information. We designed a benchmark experiment in which we trained two identical NNs with two different datasets. One NN was trained with the dataset extended by our new proposed approach and one with the dataset extended by an established approach. For both datasets, we used the same original data and parameters. Compared to current approaches, the NNs trained with our workflow augmented data achieve a better classification accuracy than the comparison network. Based on our preliminary results, we believe that our proposed approach has a high potential to improve video classification and recognition not only in the medical field. The approach will be validated in future work, and its potential in other fields like gait analysis or further applications will be shown. In addition, the workflow model could be extended, and refined by other information, e.g., anatomical data or special patient data, because physicians often use data on patient history, age, demographics for the decision-making process. Finally, we were able to show that it is advantageous to build a workflow model from annotated data. The synthetic data created with the workflow augmentation led to better classification results by NNs.

Chapter 5

Improving Surgical Phase Recognition in Videos using Workflow Augmentation

5.1 Introduction

Automatic recognition of surgical processes and workflows will be a key feature of future intelligent context-aware operating rooms [84, 85]. The aim is to improve the quality and safety of surgical treatment [86]. For this purpose, automatic context-aware systems are a mandatory prerequisite for intelligent surgical assistance systems [87]. Such a system can automatically monitor the procedures and surgical intervention, thus early alerting possible anomalies and discrepancies [88–91] or improving personnel planning [92, 93] and the coordination in the surgical team [94]. Retrospective workflow analysis of videos could improve the surgical skills assessments, the automated documentation of surgical reports, surgeon training, or postoperative patient monitoring [17, 21, 95].

Computational surgery and artificial intelligence can automatically extract workflow steps, also called phases, from videos. However, pure video-based workflow recognition is very complex because surgical scenes usually have only a small, limited variance between adjacent phases, but they have a significant variance within a phase [96]. Various attempts have been made to recognize the individual phases also by other information such as binary instrument usage signals [17], radio-frequency identification (RFID) tags [97], or by special sensors on tool tracking devices [98]. However, these techniques all have fundamental drawbacks. They require either error-prone human interaction or additional equipment for detection, which can affect the established workflow in operation and, thereby, increasing the surgical risks.

However, video recording systems are a well-established approach to document surgery in clinical practice. Concerning minimally surgical interventions, microscopes or endoscopic systems often use a video streaming system to record parts of the operation. Therefore, phase detection based on videos, which were acquired routinely during the surgical procedure [21, 89, 90], provides an easy and inexpensive concept for the operating room.

Several approaches have been developed to automatically detect surgical phases based on endoscopic, or more precisely, laparoscopic video data, in recent years. Until 2015, the hidden Markov model (HMM) and dynamic time warping (DTW) were mainly used [17, 99–105]. The accurate recognition rate of the surgical phases was less than 90 %. In the following years, improvements of the recognition rate up to 92.8 % could be achieved with the help of a combined convolutional neural network (CNN) and HMM or DTW [106, 107]. Recently, recurrent neural networks (RNNs) outperformed the previous methods with a recognition rate of up to 96.3 % [108–112]. In most cases, first, a CNN is used for feature extraction, but instead of rigid HMMs or DTWs, RNNs are used to model the temporal relationships. The advantage of using RNNs is that they can adaptively learn the individual phase transitions based on the given labels. However, there are major disadvantages to using RNNs. First, labels must be annotated before, and second, RNNs require a large amount of training data [112].

For enlargement the trainings dataset, the videos can be augmented in the image domain using traditional methods from computer vision. These methods include morphological function, e.g., rotation, shifting, padding, ..., but also functions the change the contrast, brightness or the signal-to-noise ratio. However, these methods do not increase temporal or temporal structural variability, which can be very important for the RNNs to generalize well in clinical application [113].

By using our novel approach, which we have presented in project *Workflow Augmentation of Video Data for Event Recognition with Time-Sensitive Neural Networks*, described in Chapter 4, it is possible to augment beside the image domain the temporal domain also. This approach allows the creation of new artificial laparoscopic videos. Additionally, it is possible to increase the balance of phase transition and the original tool variability within a phase of an existing dataset. We hypothesize that the recognition rate of a neural network (NN) based on CNNs and RNNs will be increased compared to the literature if the training dataset has been previously augmented by the workflow augmentation method.

5.2 Methods

5.2.1 Database

This study aims to evaluate the potential of our workflow augmentation method for extending and balancing the surgical phase transitions and simultaneously to increase variability within the phases of a given dataset. Therefore, we chose the Cholec80 dataset [21]. Compared to the previously used Cataract dataset [22], it includes frame-by-frame annotation regarding the surgical phases in addition to the surgical tools. The Cholec80 dataset consists of 80 videos of cholecystectomy surgeries performed by 13 surgeons at the University Hospital of Strasbourg. The videos were recorded with 25 fps. The recording resolution was mostly 854×480 pixels, except for some recordings with a resolution of 1920×1080 pixels. A senior surgeon annotates the surgical phase and the surgical tools. For each frame, a phase label was provided. The surgical phases are P1: *preparation*, P2: *calot triangle dissection*, P3: *clipping cutting*, P4: *gallbladder dissection*, P5: *gallbladder packaging*, P6: *cleaning coagulation*, P7: *gallbladder retraction*. Figure 5.1 shows the sequence graph with transition probabilities of the phases from the Cholec80 dataset. The graph shows also in green the start phases and in orange the terminal phases.
The presented surgical tools are only labeled every 25 frames, i.e., every second. A tool was defined as present when at least half of the tooltip was visible. There are seven types of surgical tools in total, namely: *grasper*, *bipolar*, *hook*, *scissors*, *clipper*, *irrigator*, and *specimen bag*, depicted in Figure 5.2.

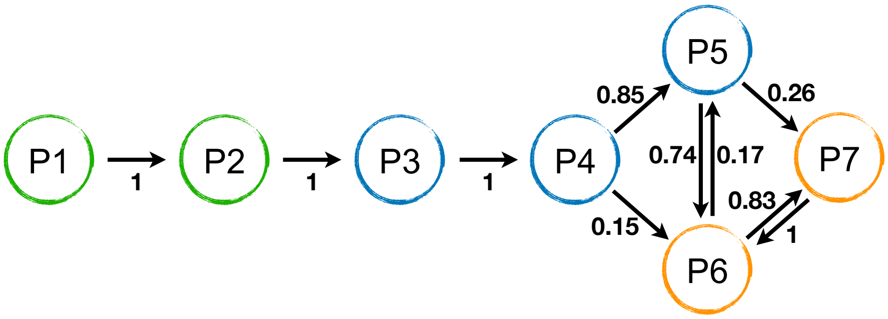


Figure 5.1: Phases transition distribution of the original Cholec80 dataset. The start phases are shown in green and in orange the terminal phases. P1: *preparation*, P2: *calot triangle dissection*, P3: *clipping cutting*, P4: *gallbladder dissection*, P5: *gallbladder packaging*, P6: *cleaning coagulation*, P7: *gallbladder retraction*

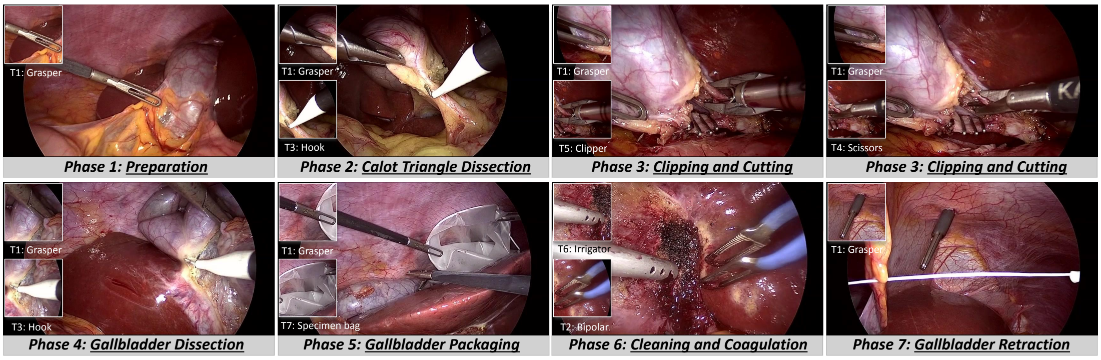

T1: Grasper
Phase 1: Preparation
T1: Grasper
T3: Hook
Phase 2: Calot Triangle Dissection
T1: Grasper
T5: Clipper
Phase 3: Clipping and Cutting
T1: Grasper
T4: Scissors
Phase 3: Clipping and Cutting
T1: Grasper
T3: Hook
Phase 4: Gallbladder Dissection
T1: Grasper
T7: Specimen bag
Phase 5: Gallbladder Packaging
T6: Irrigator
T2: Bipolar
Phase 6: Cleaning and Coagulation
T1: Grasper
Phase 7: Gallbladder Retraction

Figure 5.2: An overview of some presented surgical tools and all surgical phases from the Cholec80 dataset. Adapted from [114]

5.2.2 Data Augmentation

We describe the data augmentation pipeline and the parameter for the creation of the training dataset in the following. Therefore, we used 60 out of the 80 videos for the training dataset, the remaining 20 videos (*1*, *7*, *8*, *10*, *15*, *18*, *19*, *23*, *24*, *33*, *35*, *39*, *41*, *42*, *47*, *57*, *64*, *69*, *72*, *74*) were kept for later testing purposes and were therefore excluded from all data augmentation steps. The video selection for the training and test dataset was performed randomly.

5.2.2.1 Extraction of Required Components for Workflow Augmentation

For the data augmentation, the following components are required: the transition matrix of phases in combination with surgical tools, the spatial augmentation parameters, and the temporal augmentation parameters. Next, these components are described in more detail.

For the automatic extraction of the transition matrix, we used the original annotations for the phases and the surgical instruments. Since the surgical tools are only available for every 25th video frame, we used nearest-neighbor interpolation for the intermediate 25 frames before and after a labeled time step. The entries of the transition matrix contain only transitions between two events. An event is defined by the unique phase and, if presented, at least one surgical tool. Furthermore, the beginning and end of an event are defined by the change of annotated phase or the annotated tools. This results in 60 individual events. Afterwards, we performed a row-wise normalization of the transition matrix such that the entries no longer represent the event transitions but instead represent their probabilities. To balance the phase transitions after augmentation, resulting in a more uniform distribution than in the original dataset, we have to adjust the corresponding entries per line that implies a phase transition. The corresponding entries are adjusted to equal the desired probability per line of the sum of phase transitions from a phase p_i to a phase p_j. For this purpose, the probabilities of all phase transitions are summed up and divided according to their occurrence. The adjusted event entries E for the phase transitions $i \rightarrow j$ are defined as follows:

$$e_{(i \rightarrow j)_{new}} = \frac{\left\| \sum_{i,j=0;j \neq i}^{P} \sum_{k=0}^{E} e_{k_{(i \rightarrow j)_{old}}} \right\|}{\left| e_{k_{(i \rightarrow j)}} \right|} \tag{5.1}$$

Here, $P = 7$ denotes the number of different phases, E the number of event entries, and i, j is the dedicated phases transitions. The other transition matrix entries are not changed because this could strongly influence the length of the artificial videos. One of our goals was to create videos according to the original length artificially. At the same time, the segment database was created while creating the event transition matrix. The original videos were cut according to these events. The cut was made in the middle between two events, as described in Section 4.3.2.1. The segments consist only one dedicated event transition. The transition matrix and the segment database are the required ingredients for workflow augmentation. In addition, the parameters for the variation in appearance and time must be defined.

5.2.2.2 Spatial Augmentation Parameter

For this study, we chose only functions that varied the spatial image information, the brightness or the signal-to-noise ratio. Therefore, we use the following 15 different augmentation types shown in Table 5.1 of the batch generator of Isensee et al. [80]. We did not augment the color information because, in cholecystectomy, the color, e.g., case of spontaneous hemorrhage, can contain encoded information that also results in the use of a specific instrument, e.g., in the case of hemorrhage, the bipolar forceps in the dataset called bipolar. Each of the 15 functions is chosen randomly and uniformly distributed with a probability of 33 % afterwards, the execution order randomly shuffled due to higher appearance variability. The parameters for the specific augmentation function can be taken from Table 5.1. The values were determined empirically for the dataset.

5.2.2.3 Temporal Augmentation Parameter

For the temporal augmentation range, we chose the same 20 different factors for speed variation, as described in Section 4.3.2.1. These are divided equally distributed between $0.5\times - 1\times - 2\times$ and allow us to create videos that have the same distribution in duration as the original dataset. The concrete speed factors are the following: 2, 1.8824, 1.778, 1.6842, 1.6, 1.4884, 1.3913, 1.3061, 1.2075, 1.1637, 1, 0.9552, 0.9014, 0.8533, 0.8, 0.7529, 0.7033, 0.6534, 0.5981, 0.5517 and 0.5. These factors correspond to the following sub-frames: 128, 116, 107, 98, 91, 85, 80, 75, 71, 67, 64, 58, 53, 49, 46, 43, 40, 38, 36, 34 and 32. Therefore, the entire segment database was up-sampled by factor 64. For the annotation of the sub-frames $[1, 32]$, the labels of the original frame n were taken, and for the sub-frames $[33, 64]$, the labels of the frame $n + 1$.

5.2.3 Generating new Artificial Cholecystectomy Videos

We generated 1000 new artificial videos. Thereby, the new artificial videos are generated with the help of the Markov chain. The Markov chain is created using the transition matrix. A starting point is one of the *start events* in the transition matrix. These *start events* are equal to the start events in the original dataset. Starting from this event, the next event randomly selects, according to the transition probability, from the possible events that follow the current event in the transition matrix. This procedure is continued until a terminal event is reached.

As a result, we got 1000 different scripts for the new event sequences. To create a concrete video for each script, it is necessary that for every event one segment is randomly selected from the possible event segments in the database. All selected segments are directly attached to each other. As a result, we get a raw video sequence containing new inner phase variation. Since the complete database was up-sampled by a factor of 64 compared to the original video the raw videos are now longer and have no variation in appearance and duration. For the temporal variation, we also used a two-dimensional Halton sequence as described in Section 4.3.2.1. We linearly interpolated the samples such that the first dimension represents mean length of the original event segments and the second dimension the speed factor. The concrete values are randomly selected from the Halton sequence. The frames of the raw videos are now selected over an interval coded by the first dimension and the corresponding frame for the speed coded by the second dimension. After reaching the end of the interval, a new Halton sequence is randomly picked. The process is continued until the complete raw video is temporally augmented. Then the spatial augmentation is applied to the videos, according to the

Table 5.1: Specific augmentation parameter range for the different functions

function	min	max
center cropping	(380,480)	(804,854)
padding	(480,720)	(854,1281)
90 degree rotation	1	3
mirror axis	x,y	
zoom factor	0.85	1
random rotation	-90	90
contrast	0.2	2
additive brightness	-64	64
multiplicative brightness	0.5	1.5
gamma spreading	0.5	1.5
linear down-sampling	0.75	1.25
Rician noise	0	10
Gaussian noise	0	10
Gaussian blur	0	3.5
square noise	(0,32)	(0,150)

rule described above, see Section 5.2.2.2. As a result, the final dataset contains 1000 new artificial videos with a balanced number of phase transitions and a high inner phase variability. The 1000 videos should allow us an end-to-end training of the NN.

5.2.4 Implementation Details of the Neural Networks

To test our hypothesis and to be able to compare our results better with the literature, we chose the end-to-end NN framework proposed by Jin et al. [109] presented in Figure 5.3. They used it for the tool and phase recognition for the Cholec80 videos. The NN consists of a CNN followed by an RNN. Thereby, the idea is that the CNN extracts the feature map of the single frames, and the RNN performs temporal modeling using the feature maps. For the CNN, they chose the ResNet50 [71] and removed the output layer, and for the RNN, a layer of 512 long short-term memory (LSTM) blocks [33] and append a fully connected linear layer with seven nodes according to the surgical phases. The ResNet was pre-trained on the ImageNet dataset [115]. Afterwards, the LSTM and the linear layer weights were initialized randomly.

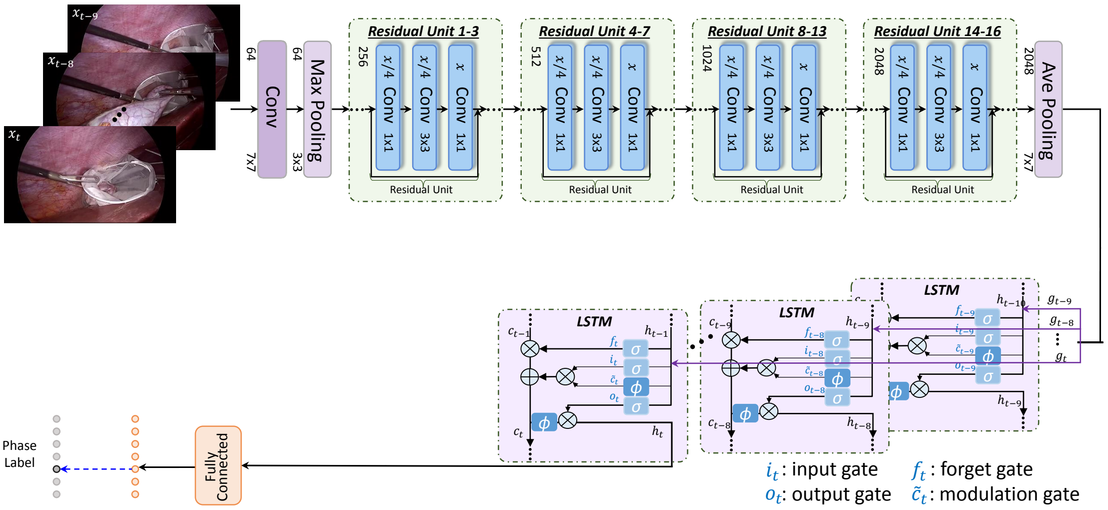


Figure 5.3: An overview of the used neural network for phase recognition from surgical videos in an end-to-end framework, modified from [114]

5.2.5 Training Strategy

For the training of the NN, the augmented datasets that contain 1000 artificial videos were randomly divided into 800 training videos and 200 validation videos. Furthermore, we normalized the frames over the complete dataset. We used the adaptive moment estimation (ADAM) optimizer with a learning rate of $l_r = 0.00001$ and the cross-entropy loss. The learning rate was constant over the training. The model has trained on one NVIDIA RTX2080Ti with 11GB memory. Due to memory constraints, a maximum of 2400 frames per video and a batch size of 100 frames are selected in chronological order. The video order of the training datasets was shuffled for each training epoch. In addition, the starting point of the video was randomly selected in the range of the first 50 frames of the 2400 selected frames. The goal was to achieve an additional variation at the phase boundaries in the batches.

5.3 Results

The goal of the work was to improve semantic information recognition performance by applying workflow augmentation. For this purpose, new artificial videos base on the existing Cholec80 dataset were augmented, and the semantic information, phases of the surgery, were balanced. The intra-phase variability should also be increased without changing the overall length of the videos. Finally, an experiment was designed and conducted to evaluate the performance of the workflow augmentation approach. The experiment addresses the classification performance of the combined CNN and LSTM network trained on the workflow augmented dataset. Before evaluating the experiment, the created training dataset was compared to the original dataset.

5.3.1 Evaluation of the Augmented Dataset

The augmented dataset contained 1000 artificial videos. They were created using the automatically extracted and modified original workflow with the 473 segment bins according to the observed events. For the workflow and segment bins, 60 of the 80 original Cholec80 videos were used, as mentioned in Section 5.2.2.

Figure 5.4 compares the original and augmented dataset's total video length. The original videos had a median total frame number of 52376 with the 25th percentile of 41026 frames and the 75th percentile of 72051 frames. The videos of the augmented dataset had a median length of 53009 frames, with the 25th percentile of 35026.5 frames and the 75th percentile of 76273 frames. The median length of the videos of the augmented dataset are compared to the original dataset in the same range. The percentile distance is larger due the 6000 frames lower 25th percentile and the 4000 frames higher 75th percentile of the augmented dataset compares to the original dataset.

In contrast, most phases in the augmented videos became shorter in median than the original videos, as seen in Table 5.2 and Figure 5.5, except P3: *clipping cutting*, which became longer to 177 frames. The phases became up to 88 % shorter (P6: *cleaning coagulation*). The table also shows that the inter-quantile range increased for all phases. On the other hand, the intra-phase variability, i.e., the change of events, increased up to

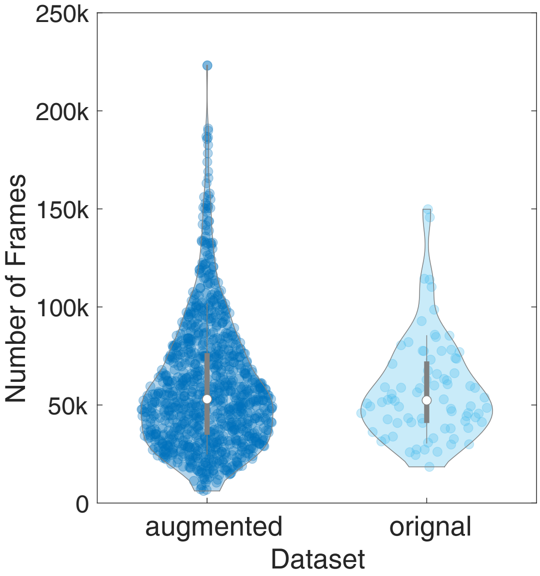
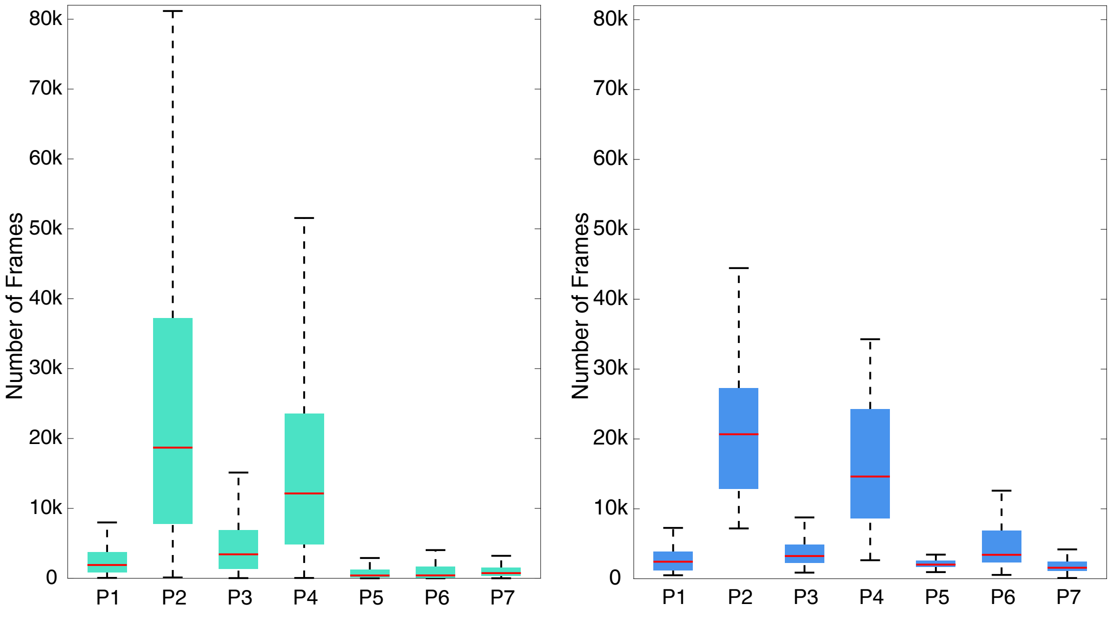


Figure 5.4: Violin plot of the total video length in frames, for the augmented and the original Cholec80 dataset.

89.6 % (P5: *gallbladder packaging*) in the augmented dataset compared to the original dataset (Figure 5.6).

Table 5.2: Comparison of phase length in frame numbers between the original and augmented dataset

Phase	dataset	25th perc.	median	75th perc.
P1:	original	1175	2437.5	3875
preparation	augmented	853	1899	3727
P2:	original	12905.25	20674	27292.75
calot triangle dissection	augmented	7762	18695	37228.25
P3:	original	2311.5	3249	4911.5
clipping cutting	augmented	1347.75	3426	6874
P4:	original	8605.25	14624	24236.5
gallbladder dissection	augmented	4856.25	12138	23544.25
P5:	original	1705.25	2024	2567.75
gallbladder packaging	augmented	131	399	1241
P6:	original	2324	3424	6836.5
cleaning coagulation	augmented	75	411	1666
P7:	original	1106.25	1575	2437.5
gallbladder retraction	augmented	330.5	736	1486



(a) Augmented dataset

(b) Original dataset

Figure 5.5: Boxplot for the different phase lengths in frames from **(a)** the augmented dataset and **(b)** the original Cholec80. (P1: *preparation*, P2: *calot triangle dissection*, P3: *clipping cutting*, P4: *gallbladder dissection*, P5: *gallbladder packaging*, P6: *cleaning coagulation*, P7: *gallbladder retraction*)

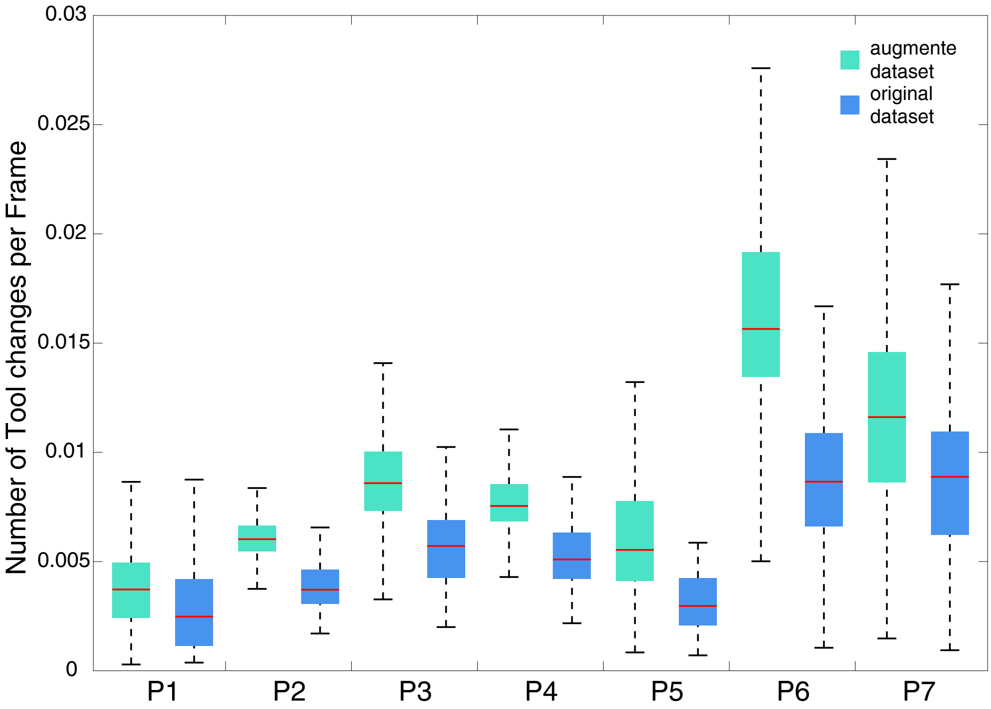


Figure 5.6: Boxplot for the tool changes per frames of the different phases from the augmented dataset (green) and the original Cholec80 dataset (blue). (P1: *preparation*, P2: *calot triangle dissection*, P3: *clipping cutting*, P4: *gallbladder dissection*, P5: *gallbladder packaging*, P6: *cleaning coagulation*, P7: *gallbladder retraction*)

Table 5.3: Quantitative phase transition of **(a)** the augmented 1000 videos and **(b)** the original 80 videos from the Cholec80 dataset. (P1: *preparation*, P2: *calot triangle dissection*, P3: *clipping cutting*, P4: *gallbladder dissection*, P5: *gallbladder packaging*, P6: *cleaning coagulation*, P7: *gallbladder retraction*)

(a) Original dataset

	P1	P2	P3	P4	P5	P6	P7
P1	0	71	0	0	0	0	0
P2	0	0	80	0	0	0	0
P3	0	0	0	80	0	0	0
P4	0	0	0	0	68	12	0
P5	0	0	0	0	0	59	21
P6	0	0	0	0	12	0	59
P7	0	0	0	0	0	3	0

(b) Augmented dataset

	P1	P2	P3	P4	P5	P6	P7
P1	0	655	0	0	0	0	0
P2	0	0	1000	0	0	0	0
P3	0	0	0	1000	0	0	0
P4	0	0	0	0	571	429	0
P5	0	0	0	0	0	2021	547
P6	0	0	0	0	1997	0	1678
P7	0	0	0	0	0	1270	0

Comparing the graph of phase transitions of the original dataset in Figure 5.1 with the graph of phase transitions of the augmented dataset in Figure 5.7, it gets apparent that the probabilities for the transition, which correspond to the probabilities of the outgoing edges, were more uniformly than in the original dataset. However, there was one exception the outgoing edges of P5: *gallbladder packaging*. The phase transitions for P5 of the augmented dataset was with 0.79 (P5→P6) and 0.21 (P5→ P7), almost identical with 0.74 (P5→P6) and 0.26 (P5→ P7) to those of the original dataset. Table 5.3 additionally shows that the absolute number of transitions from P7: *gallbladder retraction* to P6: *cleaning coagulation* has increased from 3 in the original dataset to 1270.

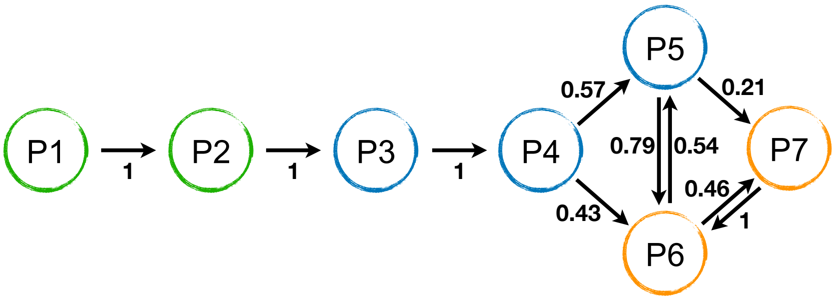


Figure 5.7: Phases transition distribution of the augmented 1000 videos. The start phases are shown in green and the terminal phases in orange. (P1: *preparation*, P2: *calot triangle dissection*, P3: *clipping cutting*, P4: *gallbladder dissection*, P5: *gallbladder packaging*, P6: *cleaning coagulation*, P7: *gallbladder retraction*)

5.3.2 Phase Recognition

The evaluation of the phase recognition performance was done with the NN proposed by Jin et al. [109] described in Section 5.2.4. The NN demonstrated its capability to detect the phases in the surgical videos by being the best NN at the M2CAI 2016 challenge

[116]. For the experiment, the NN was trained (details in Section 5.2.5) on the workflow augmented dataset, and then the phases recognition performance was evaluated on 20 videos that were not used to create the augmented dataset. Due to time constraints, the training was only performed over 20 epochs, and it did not fully converge yet. The training of the NN ends with a validation misclassification loss of 1.1^{-6} and an accuracy (ACC) of 98.15 %. A training epoch takes about 4.5 hours for the NN on a NVIDIA RTX2080Ti.

For the evaluation of the classifiers, we chose a sequence of a maximum of 2400 video frames due to memory constraints. The following overall metrics for multi-class imbalanced datasets were carried out [83]: accuracy (ACC), mean accuracy (AvACC), class balanced accuracy (CBA), macro-mean recall (REC_M), macro-mean precision (PREC_M), and macro F1-score (F1-score_M) for the network. The classifier predicted the correct phases with an average ACC of 97.0 % over the 20 videos and an overall frames ACC of 89.36 %. The AvACC of the test videos was 86.06 %. The CBA was 82.95 %. The classifier hits the phase with a PREC_M of 84.22 % correctly. The REC_M was 86.24 %, and the F1-score_M was 85.22 %.

Table 5.4 shows the accuracy (ACC), PREC, REC, SPEC, and F1-scores for the binary analysis for each phase. The respective phase was evaluated against all others, aggregated into one. In other words, the table shows the classifier's ability to recognize a concrete surgical phase. Thereby, P4 was predicted the worst with an ACC of 93.59 % by the NN compared to the other phases. P3 had the lowest PREC of all phases with 72.33 %. P5, with 77.97 %, had the lowest REC of the phases. The SPEC was over 96.25 % for every phase. The F1-score was lowest for P5 with 78 %.

The confusion matrix is presented in Table 5.5 of the NN for the selected frames of the test dataset. As it can be seen, the labeled phases were mainly confused with their neighboring phases, except P4, where a wrong prediction was distributed over all phases. It is also noticeable that for P1 to P3, only the phases P1 to P4 were predicted, and for phases P5 to P7 the phases P4 to P7 were predicted.

Table 5.4: Binary analysis of the phases for NN on the test video of the Cholec80 dataset with the scores for accuracy (ACC), precision (precision (PREC)), recall (recall (REC)), specificity (specificity (SPEC)), and F1-score, rounded on 4 digits.

Phase	ACC	PREC	REC	SPEC	F1-score
P1: *preparation*	0.9922	0.9015	0.9207	0.9955	0.911
P2: *calot triangle dissection*	0.9629	0.937	0.9636	0.9625	0.9501
P3: *clipping cutting*	0.9634	0.7233	0.881	0.9706	0.7944
P4: *gallbladder dissection*	0.9359	0.9452	0.8622	0.9741	0.9018
P5: *gallbladder packaging*	0.9791	0.7852	0.7797	0.9892	0.7824
P6: *cleaning coagulation*	0.9714	0.812	0.812	0.9845	0.812
P7: *gallbladder retraction*	0.9824	0.7908	0.818	0.990	0.8042

Table 5.5: Confusion matrix rounded to 4 digits of the selected real test videos from the Cholec80 dataset for the NN

Prediction **Label**	P1: *preparation*	P2: *calot triangle dissection*	P3: *clipping cutting*	P4: *gallbladder dissection*	P5: *gallbladder packaging*	P6: *cleaning coagulation*	P7: *gallbladder retraction*	Recall
P1: *preparation*	1858 0.9207	160 0.0793	0	0	0	0	0	0.9207
P2: *calot triangle dissection*	194 0.0113	16523 0.9636	278 0.0162	153 0.0089	0	0	0	0.9636
P3: *clipping cutting*	1 0.0003	90 0.024	3309 0.881	356 0.0948	0	0	0	0.881
P4: *gallbladder dissection*	8 0.0005	860 0.0539	988 0.0619	13767 0.8622	192 0.012	140 0.0088	12 0.0008	0.8622
P5: *gallbladder packaging*	0	0	0	116 0.0515	1755 0.7797	285 0.1266	95 0.0422	0.7797
P6: *cleaning coagulation*	0	0	0	173 0.0486	156 0.0438	2893 0.812	341 0.0957	0.812
P7: *gallbladder retraction*	0	0	0	0	132 0.0637	245 0.1183	1694 0.818	0.818
Precision	0.9015	0.937	0.7233	0.9452	0.7852	0.8120	0.7908	

5.4 Discussion

One goal of this work was to create new artificial laparoscopy videos while achieving a balance of original phase transition and higher tool variability within a phase compared to the original dataset. The hypothesis was that the recognition rate of CNN and RNN combination would increase compared to the literature approaches when trained on a training dataset augmented by the workflow method. For this purpose, we selected the laparoscopic dataset Cholec80 as an example. The dataset contains 80 surgical video recordings with the individual phases annotated.

Section 5.3.1 shows that the augmented training dataset videos have a similar median duration of approx. 53000 frames. A similar duration was desired when creating and adapting the transition matrix in Section 5.2.2.1. However, as seen in Figure 5.4, there were also videos whose length was about four times longer than the median video length. These longer videos are not a problem because these are only a few outliers, and most of the 1000 videos have a length corresponding to the original dataset's distribution.

The first goal of balancing the phase transitions was only partially achieved. The outgoing edges of P4 and P6 are balanced, but the outgoing edges of P5 correspond to the original distribution. In the analysis, it turned out that the reason for the unbalanced edges was the procedure for balancing the phase transitions. It is not enough to adjust the phase transitions by an isolated adjustment and the individual modeling of the event transitions. It is important to normalize all different phase transitions of an event and a second step to change the probabilities that the phase transitions are uniform. The procedure has been demonstrated to be suitable because the probability distribution of the outgoing edges could be improved for the majority of cases.

The second goal of increasing intra-phase variability was markedly achieved, as shown in Figure 5.6. However, the variability should be expected to decrease due to the decrease of the single-phase length (see Table 5.2) and the design decision to augment the segments so that the length remains in average the same length, as described in Section 5.2.2.3.The length of the remaining augmented sections did not change the total video length, which is visible in Table 5.2. Consequently, the median length of individual events should have become shorter if the number of events stayed the same, but it became longer, as shown in Figure 5.8. The median length of individual events increased from 57 frames to 74 frames with a similar distribution. However, the shortening of the duration of the event is not harmful since the events play only an indirect role in the training. The event labels often used in a separate training of CNN and RNN are not used since we use end-to-end training by only using the phase labels and the videos.

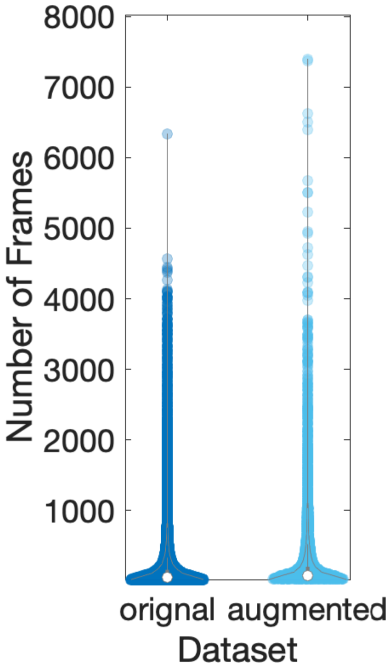


Figure 5.8: Violin plot of the event's length in frames for the augmented and the original Cholec80 dataset.

In summary, we can state that the augmentation by the workflow augmentation method has markedly enhanced the training dataset of the original Cholec80 dataset.

This raises the question, does the improvement of the training dataset also improve the phase recognition performance of the NN, as initially claimed in our hypothesis?

After only a training period of 20 epochs on 800 videos, our network reaches an ACC of 96.96 %, averaged over all test videos. The network is not yet converged, which makes the result even more impressive. Jin et al. [109] achieved only an ACC of 89.2 % overall phases. The same network trained on the same original dataset performed seven percentage points (pp) better when the workflow augmentation method is used to augment the training dataset. Compared to other publications that used the same dataset and a similar NN, our network still shows a better ACC. Yi et al., who also use a CNN and LSTM, achieved an ACC of 91.5 % [117]. Further studies and their results can be taken from Table 5.6. However, the values for PREC_M, REC_M and F1-score$_M$ from our NN are slightly lower with 84.22 % (86.9 %), 86.24 % (88 %) and 85.22 % (87.4 %) compared to Jin et al. (values in brackets).

May a closer look at the individual phase evaluations should answer why the values are slightly lower by better general ACC. Unfortunately, the values for ACC, PREC, REC, and F1-score for the individual phases were not published by Jin et al. ACC values for the individual phases could neither be found in the publication. Only Namazi et al. published the PREC, REC, and F1-scores for the individual phases, shown in Table 5.7, which were used for comparison in the following. Comparing the scores in Table 5.7 reveals small differences for the individual scores for P1 to P4. The PREC and REC for P5 in our NN are about six pp worse than in Namazi et al. study, but the values for P6 and P7 are better in comparison. Namazi et al. also provides the confusion matrix for the individual phases, which shows similar patterns for the online NN as ours (Table 5.5). One possible explanation could be that there must be inconsistencies in the dataset, especially regarding P4. However, this assumption could not be further proofed.

Table 5.6: Phase recognition results found in literature using the same Cholec80 dataset and similar NN types.

Reference	Type	ACC
Jin et al. [109]	CNN+LSTM	89.2 %
Yi et al. [117]	CNN+LSTM	91.5 %
Twinanda et al. [21]	CNN+SVM	86 %
Namazi et al. [111]	CNN+LSTM	90.8 %
Funke et al. [108]	CNN+ LSTM	92.7 %
ours	CNN+ LSTM	97.0 %

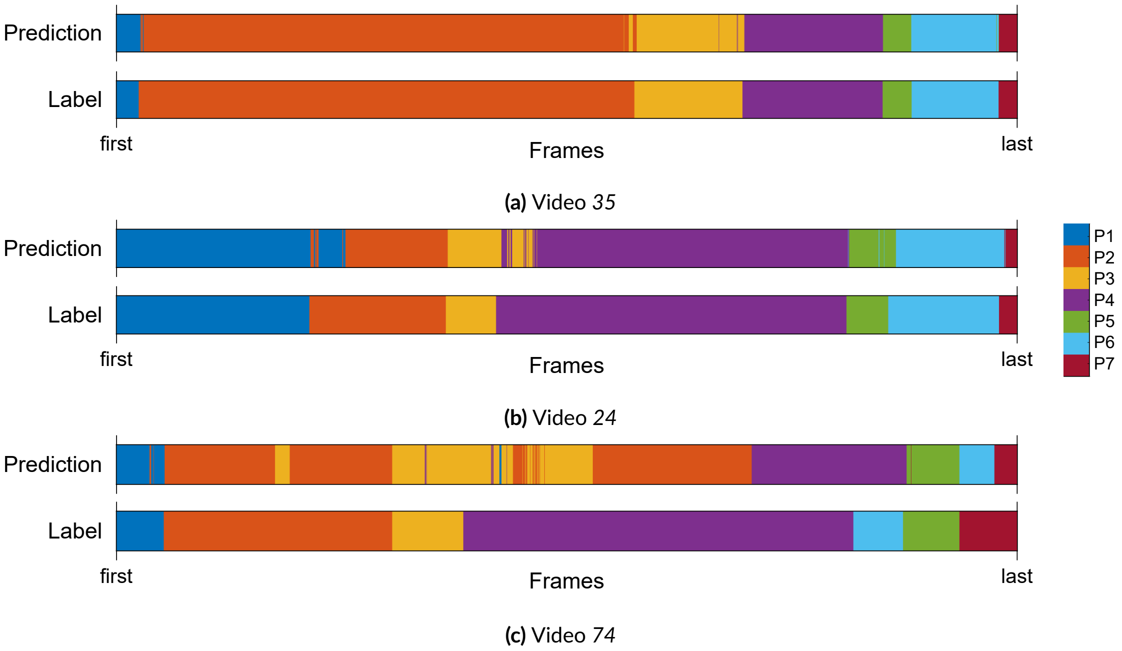
Another interesting observation was that there were also phase mismatches in the neighborhood phases of each phase. This leads to the assumption that the mismatch

Table 5.7: Comparison of scores for accuracy (ACC), precision (PREC), recall (REC), specificity (SPEC), and F1-score for the different phases, between this work and Namazi et al. [111]

Phase	Namazi et al.			**ours**		
	PREC	REC	F1-score	PREC	REC	F1-score
P1: *preparation*	0.88	0.90	0.89	0.9015	0.9207	0.911
P2: *calot triangle dissection*	0.94	0.97	0.96	0.937	0.9636	0.9501
P3: *clipping cutting*	0.72	0.81	0.76	0.7233	0.881	0.7944
P4: *gallbladder dissection*	0.96	0.92	0.94	0.9452	0.8622	0.9018
P5: *gallbladder packaging*	0.87	0.84	0.85	0.7852	0.7797	0.7824
P6: *cleaning coagulation*	0.82	0.65	0.72	0.812	0.812	0.812
P7: *gallbladder retraction*	0.65	0.74	0.69	0.7908	0.818	0.8042

occurs mainly at the transitions. We looked at three plots containing the temporal progression of prediction and ground-truth label to check this assumption, as shown in Figure 5.9. The first one is video *35* with the best ACC (98.45 %), the second video *24* with medium ACC (98.45 %), and the third video *74* with the worst ACC (56.28 %). The figure confirms the stated assumption for the first two cases. It is visible that there are inconsistencies at the phase transitions. However, for the third case, we must note that the phases at the beginning and end of the video are still correct, but large parts, in the middle, have been incorrectly ordered. However, we must emphasize again that video *74*, was the video that had by far the worst recognition ACC with 56.28 %. For all other videos the ACCs were above 83 %.

To conclude, the experiment confirmed our hypothesis, and the recognition performance could be significantly increased. Using the workflow augmentation method, it is possible to augment the semantic information, i.e., phases, in an existing dataset. Furthermore, it could demonstrate that enhancing the training dataset by the presented method increased the recognition rate significantly compared to the literature.



(a) Video *35*

(b) Video *24*

(c) Video *74*

Figure 5.9: Illustration of predicted phases compare to the ground-truth on the three test videos: *35* (best ACC), *24* (average ACC) and *74* (worst ACC) from the Cholec80 dataset. The different phases are color coded and horizontal axis represents the time progression in the surgery in frames. (P1: *preparation*, P2: *calot triangle dissection*, P3: *clipping cutting*, P4: *gallbladder dissection*, P5: *gallbladder packaging*, P6: *cleaning coagulation*, P7: *gallbladder retraction*)

5.5 Conclusion

In this project, we demonstrated that the novel approach presented for augmenting video in the field of video-based event recognition in Chapter 4 could also be used to augment semantic information, e.g., surgical phases in a video. This methodology enables the creation of new artificial videos from originally recorded videos while at the same time balancing the phase transitions. Moreover, it is possible to increase the intra-phase variability compared to the original videos. We have demonstrated that a NN, consisting of a CNN and LSTM, achieved significantly better performance in terms of surgical phase-detection, compared to literature, if it's trained on a dataset augmented by the workflow augmentation method. Moreover, our approach outperformed all other known published approaches regarding ACC with an increase in minimum of 4.5 % (see Table 5.6).

However, there are also limitations to our approach. If there are inconsistencies in a dataset, as we presume for P4, these could be eliminated by dropping the questionable events in the workflow approach, but if they are not known, then the effect of the incorrect segments could even be exacerbated by the workflow method. Therefore, the initial intervention quality of the dataset must be sufficient, i.e., the recorded videos include only professional standardized workflows by skilled physicians. Besides, the dataset should have sufficient diversity because the workflow method can increase the variability of existing events over time but not generate new events.

Based on these results, the approach proposed in Chapter 6 has a high potential to improve video classification and recognition, especially of rare events in the field of computational-assisted surgery. The better automatic recognition of the context can increase patient safety, as the surgeon can be warned at an early point that she/he does not follow the standardized procedure, or assistance can be given especially, in rare cases. With the help of workflow augmentation, a great step towards the future intelligent context-aware operating room is taken.

The next logical step of artificial data generation is to synthetic generate realistic images or videos for rare events. Such events, with low prevalence, are challenging to find in datasets and are not likely to be recorded. With the workflow extension, such data could improve sporadic events' classification or recognition performance using artificial intelligence.

Chapter 6

3D-Guided Face Manipulation of 2D Images for the Prediction of Post-Operative Outcome after Cranio-Maxillofacial Surgery

6.1 Introduction

Cranio-maxillofacial surgery is a common treatment of temporomandibular disorders or skeletal malocclusion. Besides the improvement of function, this surgical intervention often changes the aesthetics or identity of the face, which can be a heavy burden for the patient. To support the patient's decision-making in favor of or against surgery, having a prediction of the patient's face after surgery is highly desirable. At present, physicians can predict the virtual post-operative face using surgery planning tools like IPS CaseDesigner® [118] or Dolphin 3D® [119]. These surgery planning tools typically require a tomography scan of the patient's face, which includes both segmented soft-tissue and segmented bone structure. The surgery planning tool allows the physician to virtually manipulate the bone structure, e.g., to cut and move the jaw and subsequently predict the deformation of the soft tissue using, e.g., finite element methods [120–122] or mass tensor models [123]. In the next step, the texture of the face has to be predicted to allow a rendering of the post-operative face. For this, a 3D scan of the facial texture must be captured by a 3D camera system, wrapped on the virtual pre-operative face, and subsequently interpolated according to the predicted deformation of the soft-tissue [124].

This procedure to predict the post-operative texture has multiple disadvantages:

1. The procedure requires a 3D texture scanner which might not be available at every clinical site. In such a case, patients can only be provided with a single-color prediction of the post-operative face.
2. The quality of the mapped texture of the face is limited by the registration accuracy and the resolution of both the texture and the tomography scans.
3. Existing methods to translate the pre-operative texture to the predicted post-operative soft-tissue, e.g., interpolation, might be unsuitable to predict realistic textures since they do not consider illumination or skin properties.

In practice, these disadvantages often result in predictions of the post-operative face that do not look realistic or lively looking and are therefore ill-suited to support the patient's decision-making.

In this study, we propose a novel deep learning-based idea to directly predict a realistic 2D image of the post-operative face given only a 2D image of the patient before surgery and a 3D simulation of the post-operative soft tissue. In other words, we hypothesize that the current method to capture, wrap and interpolate the 3D texture can be replaced by a neural network to make a realistic prediction of the post-operative face and thus, does not require a 3D texture scanner. Our main contribution is a conditional generative adversarial network (cGAN) for post-operative face prediction which translates a 2D image of the pre-operative face of a patient to a 2D image of the post-operative face. Compared to previous approaches to predict the post-operative face [118–124], we propose a deep learning-based solution, i.e., we aim to train a suitable model directly from data. However, acquiring large numbers of corresponding image pairs between pre- and post-operative faces of cranio-maxillofacial surgery is difficult and often includes data with large time gaps of several months between images of a pair due to the long healing phase of the swelling. To bypass this lack of feasible training data, we propose a semi-supervised cycle generative adversarial network (CycleGAN) [125] strategy to train our model on non-clinical data and subsequently transfer our model to predict the post-operative face, as shown in Figure 6.1.

More precisely, we first train a modified CycleGAN on "in-the-wild" images of the 3DDFA dataset [126], where we aim to manipulate distinct local face properties of 2D images such as changing the size of the chin or the nose. In contrast to recent state-of-the-art models used to manipulate facial properties, we cannot describe the desired manipulation or a surgery plan by discrete attributes (e.g., brown/blond/black hair color in StarGAN [127, 128]), domain transfer between two images [128, 129], or concealed representations in latent space [130, 131]. Instead, we define the geometric shape of the manipulation using a 3D surface template of the face which enables precise manipulation according to the 3D shape of the face. Using this 3D template, we use a statistical model to generate distinct local face modifications and pass these locally modified 3D faces together with an unmodified 2D image to our model to predict a face that comprises the desired local manipulation. We then transfer our model to the second stage of our study,

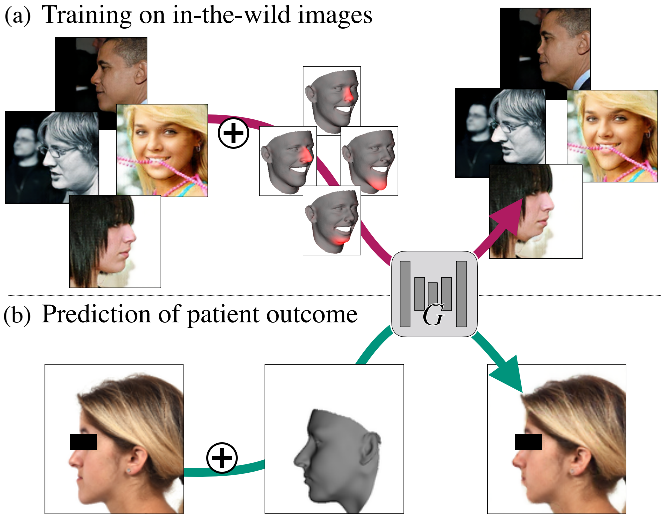

(a) Training on in-the-wild images
(b) Prediction of patient outcome
G

Figure 6.1: Overview of our approach to predict the post-operative face. In (a) we first train our model to predict various face modifications (marked in red) at the chin and the nose using a CycleGAN strategy. After training, we are able to apply local modifications to a 2D image of a face as seen on the right. Afterwards, we transfer our model to (b) where we use our trained model to predict the face of a patient after cranio-maxillofacial surgery. More precisely, we use a 2D image of the pre-operative face as shown on the left and a 3D simulation of the post-operative surface as shown in the middle to generate a prediction of the post-operative face as shown on the right. Hereby, our approach requires neither clinical data for training nor 3D texture scans for inference.

where we predict the post-operative face for two different views of four clinical subjects that underwent cranio-maxillofacial surgery. To create such a prediction, we simulate a 3D face template of the post-operative face without texture using a surgery planning tool and pass it together with an image of the pre-operative face to our model. As a result, we demonstrate the reasonability of our approach to train on non-clinical data and subsequently predict realistic 2D images of the post-operative face. Based on these promising first results, we believe that our approach has a high potential as a future tool for post-operative face prediction. Compared to previous approaches [118–124], our approach does not require 3D texture scans or registration procedures for inference, nor do we need sparsely available clinical data, physical models, or detailed surgery expertise for training.

6.2 Related Work

Our study aims to mainly contribute to the state-of-the-art in image processing for predicting the post-operative face. In the following, we describe the state-of-the-art for the prediction of the post-operative outcome as well as the state-of-the-art for manipulating 2D images of faces. Moreover, we highlight the differences of previous work compared to our study.

Post-operative face prediction. Previous research studies [122, 124, 132–134] on the prediction of the post-operative face as well as commercially available surgery planning tools [118, 119] are mainly focused on the needs of orthodontics, i.e., the planning of bone structures and the prediction of the facial soft-tissue. As a result, the prediction of the post-operative texture is often neglected or replaced by a texture of constant skin color [122, 132] which is poorly suited to guide the patient's decision whether or not to undergo surgery. On the other hand, commercial planning software such as IPS CaseDesigner® [118] or Dolphin 3D® [119] as well as Harris et al. in [133] and Premjani et al. in [135] offer the prediction of the post-operative texture based on a 3D picture of the pre-operative face. Hereby, the soft-tissue and the bone structure are typically extracted from a cone-beam computed tomography (CBCT) scan while the facial texture is captured using a 3D stereo camera system [118, 119, 124, 133, 135]. The 3D texture is then registered and wrapped on the segmented soft-tissue. However, this procedure is both time-consuming and potentially inaccurate since the registration of the texture must typically be accomplished by surface matching algorithms using manually annotated landmarks [124, 135] or manual alignment [133]. To overcome this registration problem, other studies have proposed a simultaneous data acquisition of the stereo camera scan and the CBCT scan [134, 136]. However, such stereo photogrammetry systems are expensive to acquire and rare available since they offer hardly any additional benefit for clinical diagnostics. Once the texture is wrapped on the soft-tissue of the face, available surgery planning tools allow the physician to virtually cut and move the bone structure of the face and subsequently simulate the deformation of the corresponding soft-tissue. Afterwards, the texture of the pre-operative face must be interpolated according to the simulated soft-tissue deformation to enable a rendering of the predicted post-operative face. This procedure to simulate the post-operative texture typically does not result in lively looking and realistic rendering of faces (compare, e.g., [133, 134]) since the texture quality is limited by the resolution of the stereo camera system, the resolution of the soft-tissue scan, and the registration accuracy. Additionally, the interpolation method to manipulate the pre-operative texture according to the soft-tissue deformation does not account the illumination properties of the skin or the preservation of high-frequency details, which might further reduce image quality. In contrast, we propose a generative adversarial network (GAN)-based neural network to directly manipulate a 2D image according to a 3D plan of the simulated post-operative soft-tissue. To the best of our knowledge, using neural networks to predict the post-operative face has never been proposed before. As its most important advantage, our approach neither requires the acquisition nor registration of 3D texture scans. With regard to the impressive results of recent GANs to generate and manipulate fine-detailed and realistic images of faces in high-resolution, we further hypothesize that a GAN-based approach is able to generate more realistically-looking images of the post-operative face compared to traditional approaches and therefore, might be better suited to guide the patient's decision-making before surgery.

Face manipulation of 2D images. In recent years, GANs have shown remarkable success in generating and manipulating 2D images of faces. To manipulate a face according to a desirable attribute, numerous studies have been proposed for both purely generative models and cGANs. To enable a controlled manipulation of the face, the desired manipulation has to be represented as an interpretable input to the model. To achieve this, Guan [137] and Liu et al. [130] both found representations in the input feature vector of GANs to manipulate desired properties of the face image. In contrast, He et al. (AttGAN) [131] defined the manipulation by using both discrete attributes as well as a feature vector in latent space to manipulate the facial properties of an image. Alternatively, Bao et al. [129] and Shen et al. [138] trained a cGAN to swap key facial properties between two images, which enabled manipulation of, e.g., expression, illumination, pose, wearing sunglasses, or having beards. Also, Bansal et al. (RecycleGAN) [139] proposed a CycleGAN [125] transfer facial expressions of video data from one person to another person. Closely related to this work, Choi et al. (StarGAN) [127] adapted a CycleGAN strategy and defined the manipulation information by a vector of discrete attributes like hair color, gender, or age to manipulate faces in 2D. Most recently, Choi et al. [128] released StarGAN v2 which receives both discrete attributes and style information from another image to manipulate faces in a 2D image. As seen above, all the described studies to modify faces in 2D images defined the modification information by either abstract features in latent space, information extraction by transferring properties of other images, or discrete attributes. However, none of the above studies manipulated 2D images according to a 3D plan of the face. Consequently, their approaches would be unsuitable to manipulate a face according to a precise and individual surgery plan. In contrast, we propose a model which receives a representation of a 3D surface mesh of the face to define the manipulation of the 2D face. Hereby, our representation of the face modification is continuous, easily interpretable, and enables a precise definition of the targeted geometrical shape of the face. To the best of our knowledge, such a representation to manipulate distinct properties of a face according to a 3D plan of facial shape has never been proposed before.

6.3 Methods

The goal of this study was to train a single generator G which receives a 2D image of a face and a modified 3D shape of a face as inputs. Then, the model generates a 2D image of a face that yields the desired modification as an output. More precisely, let I^n be an image of a person's face n and S^n be the corresponding estimation of the 3D shape of the same person's face as shown in Figure 6.2 (a). Subsequently, we applied a local modification S_{mod} to every unmodified 3D shape S^n in our dataset to create a locally modified 3D shape $S^n_{mod} = S^n + S_{mod}$.

As a proof-of-concept, we applied four distinct modifications S_{mod} to each face in this study: increased size of the chin, increased size of the nose, a decreased size of the chin,

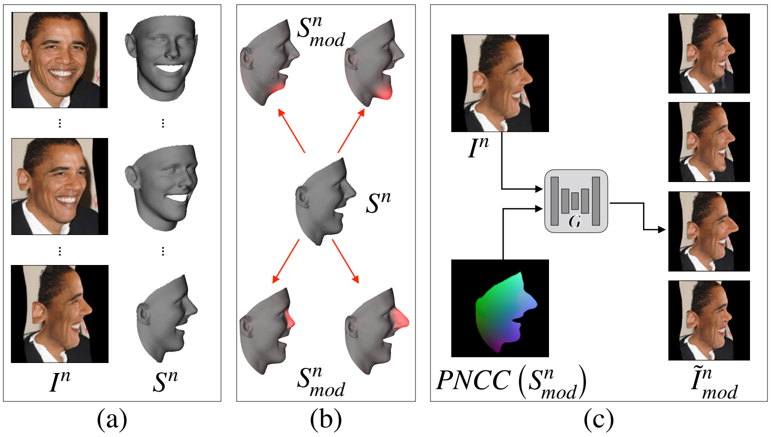


Figure 6.2: Pre-processing of the training data. (a) shows an example of the 300W-LP dataset with in-plane face rotations around the vertical axis with the angle θ by Zhu et al. [126]. Hereby, the upper image shows the original image, and the lower two images show the augmented images. Additionally, the estimated 3D shape S^n of the face was also given by the dataset. (b) shows all four face modifications S^{mod} that we applied to S^n to create the modified faces S^n_{mod} (c) shows the inputs and different outputs of the neural network G. Hereby, G received an image I^n and a projected normalized coordinate code (PNCC) projection of the modified 3D shape S^n_{mod} as input. The model then predicted the desired modification in the image $\tilde{I}^n_{mod}$.

and a decreased size of the nose as seen in Figure 6.2 (b). Technically, this approach can be extended to other deformations of the face (e.g., mouth or head modifications) as well. Next, we trained a neural network G to apply the modification described by S^n_{mod} on the original image I^n which was supposed to result in a modified image $\tilde{I}^n_{mod}$:

$$\tilde{I}^n_{mod} = G(I^n, S^n_{mod}) \tag{6.1}$$

For example, this might be an image of a face with an enlarged nose, as seen in Figure6.2 (c). To train such a model, we utilized the corresponding image I^n and 3D shape S^n pairs from the open-source “in-the-wild” 300W-LP dataset [126] and propose a semi-supervised training strategy inspired by CycleGANs [125]. Since the ground-truth of the modified faces I^n_{mod} are unknown for these “in-the-wild” images, we instead propose a training strategy that leverages four sources of a-priori knowledge to formulate our training objective:

1. a reconstruction loss as introduced by Zhu et al. [125]
2. knowledge of the statistics of real-world images of faces via an adversarial discriminator,
3. a learned mapping to translate a 2D image to a 3D shape of a face, and
4. information of the approximate location of the local modification in the image.

In the following, the applied local modification S_{mod} and the objectives for training are described in more detail.

6.3.1 Local Face Modifications

The ultimate incentive of training G was to create a model which ultimately can predict a 2D image of a patient's face after cranio-maxillofacial surgery. Generally, cranio-maxillofacial surgery does not only affect the appearance of the jaw but other regions of the face as well, e.g., the nose and the mouth, for the treatment of cleft palates. Therefore, we aimed to demonstrate that our model can be trained on arbitrary regions of the face. In this preliminary study, we chose to train our model on size variations of the nose and the chin. These local regions of the face have the advantage that they are easily recognizable in almost all images of faces and in most head positions. For training, the applied modification was required to be automatically applicable and physically plausible, i.e., that the existence of S^n_{mod} in the real world was theoretically possible. To achieve this, we expressed each 3D face S^n and each local modification S_{mod} as a parameter vector of the BFM2009 [140] statistical point distribution model. To find such local modifications S_{mod}, we annotated different local regions of the 3D face template of the BFM2009. We then implemented an optimization algorithm to find parameter vectors that result in a maximal deformation of the annotated region while minimally deflecting all other regions of the face (see Appendix B.1 for more details). Next, we scaled the computed parameter vectors in a negative and positive direction until the deformation of the desired region was maximally deflected without being unrealistic as judged by subjective inspection. The resulting four local deformations S_{mod} can be seen in Figure 6.2 (c). During training, we randomly drew one of these four modifications for each sample and applied it to the estimated 3D shape of each unmodified shape S^n of the dataset to create a modified 3D face S^n_{mod}:

$$S^n_{mod} = S^n + S_{mod} \tag{6.2}$$

6.3.2 Objectives

Image reconstruction loss. To enforce the preservation of the identity of the face in $\tilde{I}^n_{mod}$, we minimized the "identity reconstruction" loss $\mathcal{L}_{I-rec}$ where we aimed to reconstruct the original image I^n from the predicted modified image $\tilde{I}^n_{mod}$:

$$\mathcal{L}_{I-rec} = \mathbb{E}_{I^n, \tilde{I}^n_{mod}, S^n}\left[\mathcal{L}_{Perceptual}\left(I^n, G\left(\tilde{I}^n_{mod}, S^n\right)\right)\right] \tag{6.3}$$

As seen in the equation, we calculated the image distance $\mathcal{L}_{Perceptual}$ to compare the original image I^n with the reconstructed image $\tilde{I}^n = G\left(\tilde{I}^n_{mod}, S^n\right)$ as illustrated in Figure 6.3. The incentive of L_{I-rec} was to ensure that G only changes the geometric shape in $\tilde{I}^n_{mod}$ without modifying properties like skin color, facial hair, or other facial attributes independent from the facial shape that are required to translate back to the original image I^n. Consequently, these independent shape properties would have to be present in $\tilde{I}^n_{mod}$ to achieve a perfect reconstruction score. As stated in the original CycleGAN paper [125], Zhu et al. struggled to translate between images that required geometric changes (e.g., translate dogs to cats). As a possible solution, Gokaslan et al. [141] proposed the use of a perceptual loss instead of a pixel-wise loss and achieved convincing results to translate between geometrically changing images using CycleGANs. Motivated by these results, we also used a perceptual loss $\mathcal{L}_{Perceptual}$ [142] to compare between I^n and $\tilde{I}^n$. During training, we had to prevent G from learning an "arranged" encoding of these properties in $\tilde{I}^n_{mod}$ and learning a specialized decoder to reconstruct $\tilde{I}^n$. To impede the learning of such an arranged encoding, we predicted $\tilde{I}^n_{mod}$ with G and then froze the weights of G for the reconstruction of the original image $\tilde{I}^n$, i.e., all gradients induced by the second forward pass were not considered for updating G as indicated in Figure 6.3.

Shape reconstruction loss. To enforce a face manipulation in $\tilde{I}^n_{mod}$, we aimed to reconstruct the modified input shape S^n_{mod} from $\tilde{I}^n_{mod}$. For this, we first trained another neural network G_S to predict the 3D shape of an image: $\tilde{S}^n = G_S(I^n)$. After training G_S, we froze the model weights of G_S and optimized the weights of G to minimize the distance $\mathcal{L}_{S-rec}$ between the input modification S^n_{mod} and the reconstructed shape prediction $\tilde{S}^n_{mod} = G_S(\tilde{I}^n_{mod})$:

$$\mathcal{L}_{S-rec} = \mathbb{E}_{S^n_{mod}, \tilde{I}^n_{mod}} \left[\mathcal{L}_{Shape} \left(S^n_{mod}, G_S \left(\tilde{I}^n_{mod} \right) \right) \right] \tag{6.4}$$

with $\mathcal{L}_{Shape}$ being a distance metric (described below) between S^n_{mod} and $G_S(\tilde{I}^n_{mod})$. To calculate $\mathcal{L}_{S-rec}$, we first trained G_S to reach convergence using the image-shape pairs (I^n, S^n) of the 300W-LP dataset [126] and minimized the prediction error $\mathcal{L}_{G_S}$ in the shape domain:

$$\mathcal{L}_{G_S} = \mathbb{E}_{I^n, S^n} \left[\mathcal{L}_{Shape} \left(S^n, G_S \left(I^n \right) \right) \right] \tag{6.5}$$

Hereby, we assumed that the estimated 3D shape S^n of the 300W-LP dataset of each image was the ground-truth. Estimating a 3D shape of a face from a single 2D image is a highly ill-posed problem that is yet to be resolved. To solve this estimation task, current state-of-the-art studies propose either the use of iterative template fitting approaches [140, 143, 144] or regression approaches using neural networks [126, 145–148]. In initial experiments, we considered openly available algorithms or neural networks to serve as G_S and calculate $\mathcal{L}_{S-rec}$. However, we concluded that iterative algorithms [140] are unfeasible for backpropagation, and publicly available neural networks required either too much GPU RAM [145] or were locally too inaccurate [126, 147] to be used in our CycleGAN setup to calculate $\mathcal{L}_{S-rec}$. Compared to these previous studies, we aimed to facilitate the estimation task for G_S by predicting only a projection of the 3D shape. For the 3D shape, we used the PNCC proposed by Zhu et al. [126]. To calculate the PNCC, we first converted the XYZ coordinates of the mean face of the BFM2009 to the RGB color range by normalizing the coordinate range to [0, 1]. We mapped these colors on the 3D shape S^n and subsequently calculated the projection to the image plane using the projection parameters of S^n as seen in Figure 6.2 (c). By using a PNCC to represent the 3D shape of a face, we attempted to facilitate the estimation task for G_S, and we changed the translation task of our CycleGAN to a 2D problem which saved GPU RAM and enabled the use of established 2D convolutional neural network (CNN)

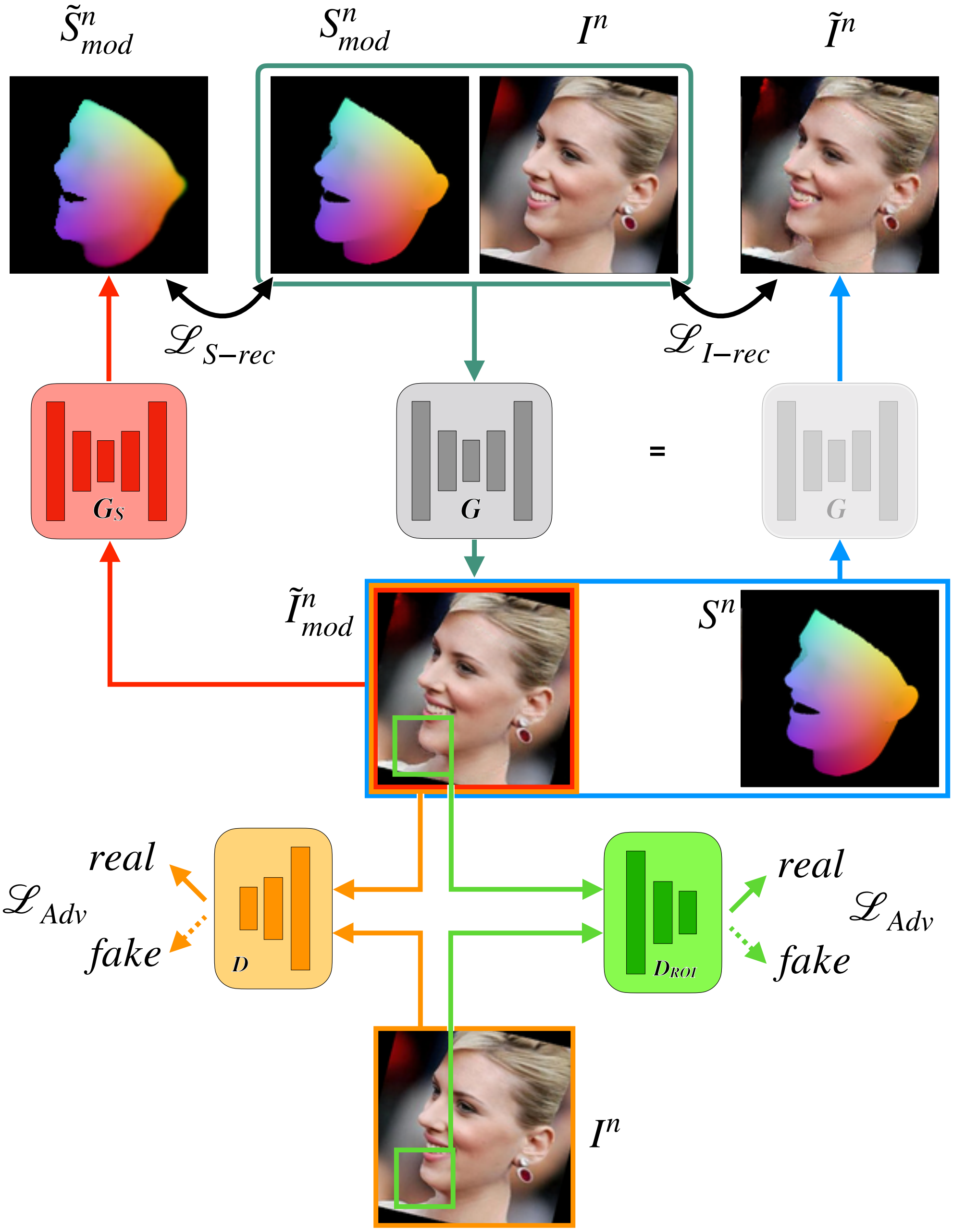


Figure 6.3: Schematic overview of our training strategy. As input, G receives an image I^n and a modified shape S^n_{mod} represented as a PNCC and predicts a modified image $\tilde{I}^n_{mod}$. To the left, G_S estimates the projected shape $\tilde{S}^n_{mod}$ from $\tilde{I}^n_{mod}$ which is then compared with the input shape S^n_{mod} to enforce the visibility of the modified shape in $\tilde{I}^n_{mod}$. On the right, G is supposed to reconstruct the original image I^n from the modified prediction $\tilde{I}^n_{mod}$ and the unmodified shape S^n. Notably, the gradients of the right-side pass were not considered for updating G during training. On the bottom, the two adversarials D and D_{Roi} aim to distinguish between generated predictions by G (fake data) and images from our dataset (real data). Hereby, D received all images at full resolution (128×128 pixels) while D_{Roi} received all images at various resolutions centered around the local modification (here: 32×32 pixels).

architectures. Notably, we also split the prediction task of G_S by separately predicting a color map of the PNCC and a mask of the PNCC, which, in practice, appeared to strongly increase convergence speed when training G_S. To calculate the distance metric between the PNCCs s $\mathcal{L}_{Shape}$, we calculated the binary cross-entropy $\mathcal{L}_{CE}$ between the masks and the L1 norm $\mathcal{L}_1$ between the color maps:

$$\begin{aligned}\mathcal{L}_{Shape}(S,\tilde{S}) =& \mathcal{L}_{CE}(S_{Mask},\tilde{S}_{Mask})\\ &+ \lambda\mathcal{L}_1(S_{Color},\tilde{S}_{Color})\end{aligned} \tag{6.6}$$

Hereby, with $\lambda = 10$ and we masked $\tilde{S}_{Color}$ with the predicted PNCC mask $\tilde{S}_{Mask}$. In the following, we name S^n or S^n_{mod} and mean the PNCC representation of the 3D face.

Adversarial loss. To restrict G to only predict realistic images $\tilde{I}^n_{mod}$, we used an adversarial loss. To calculate the adversarial loss $\mathcal{L}_{Adv}$ we randomly drew images I^n from the real data distribution $\mathcal{P}_R$ and modified images $\tilde{I}^n_{mod}$ from the fake data distribution $\mathcal{P}_G$ generated by G and random modifications S^n_{mod}. Thereafter, we alternately approximated the Wasserstein-distance by training an evaluator D and minimizing the estimated Wasserstein-distance by optimizing G according to Wasserstein generative adversarial network (WGAN) theory [38]. To enforce local 1-Lipschitz continuity in D, we adopted the gradient penalty loss (WGAN-GP) by Gulrajani et al. [149]:

$$\begin{aligned}\mathcal{L}_{Adv} =& \mathbb{E}_{I^n}\left[D\left(I^n\right)\right] - \mathbb{E}_{\tilde{I}^n_{mod}}\left[D\left(\tilde{I}^n_{mod}\right)\right]\\ &- \lambda_{GP}\mathbb{E}_{\dot{I}}\left[\left(\left\|\nabla D\left(\dot{I}\right)\right\|_2 - 1\right)^2\right]\end{aligned} \tag{6.7}$$

with $\dot{I}$ being a linear interpolation between an image pair (I^n, $\tilde{I}^n_{mod}$) and $\lambda_{GP} = 10$. Hereby, D aimed to maximize $\mathcal{L}_{Adv}$ while G aimed to minimize $\mathcal{L}_{Adv}$. To increase convergence speed and image quality, we implemented a multi-scale discriminator setup by training a second evaluator D_{Roi} on a cropped region of the local modification as seen in Figure 6.3. To automatically compute this region of interest, images I^n and $\tilde{I}^n_{mod}$ were cropped around the location of the modification, i.e., an annotated center point of the nose or the chin. To find the approximate center in the prediction of the modified face $\tilde{I}^n_{mod}$, we projected the center-point of the modified 3D shape S^n_{mod} on the image plane. During the training of G and D_{Roi}, we varied the size of these regions of interest between 16×16 pixels and 48×48 pixels before presenting them to our second evaluator D_{Roi} (see Section 6.3.4 for further details). As a result, the use of this second discriminator appeared to significantly increase convergence speed and the image quality of the predictions $\tilde{I}^n_{mod}$.

6.3.3 Datasets

For training, we used the 300W-LP dataset by Zhu et al. [126], which comprises corresponding pairs of 2D images I^n of faces, estimated parameters of the BFM2009 [140] statistical point distribution model, and projection parameters of each face. For augmentation, Zhu et al. rotated and flipped all faces in-plane around the vertical axis to comprise more training samples with high degrees of face rotations θ as seen in Figure 6.2 (a). These in-plane rotations resulted in 300 575 image and 3D shape pairs with a baseline of 7690 independent pairs. For validation, we excluded eight baseline pairs before augmentation which resulted in 112 pairs after augmentation. Additionally, we rotated each image between $-90°$ and $90°$ for further augmentation during training. Lastly, we cropped the original image resolution of 450×450 pixels around the center of the face to a resolution of 315×315 pixels and subsequently rescaled each image to a resolution of 128×128 pixels due to memory constraints of our GPU during training. For testing, we used the AFLW2000 dataset by Zhu et al. [126], which includes 2000 images, fitted 3D shapes, and projection parameters derived using the same semi-automatic template fitting approach by Paysan et al. [140] as the 300W-LP dataset.

6.3.4 Implementation Details

Training strategy. In a first step, we trained the shape estimator G_S on the 300W-LP dataset by minimizing $\mathcal{L}_{G_S}$:

$$\min_{G_S} \mathcal{L}_{G_S} \tag{6.8}$$

We used Adam [150] for optimization with $\beta_1 = 0.5$, $\beta_2 = 0.999$, a batch-size of 32, and a constant learning rate of $lr = 10^{-4}$ over the first 150 000 iterations. Then we linearly decreased lr to zero over another 150 000 iterations, which took approximately two days on an NVIDIA RTX2080Ti. After training G_S, the mean absolute pixel-wise error was $L_1 = 0.015$ and the cross-entropy loss was $L_{CE} = 0.160$ on the 300W-LP dataset and $L_1 = 0.072$ and $L_{CE} = 0.338$ on the AFLW2000 dataset. Next, we trained our CycleGAN by updating the evaluators D and D_{Roi} alternatingly on every iteration using the objective function

$$\max_{D} \mathcal{L}_{Adv} \tag{6.9}$$

while updating the generator G every fifth iteration using the objective function:

$$\begin{aligned} \min_{G} \mathcal{L}_G = &\lambda_1 \, \mathcal{L}_{I-rec} + \lambda_2 W \, \mathcal{L}_{S-rec} \\ &+ \lambda_3 \, \mathcal{L}_{Adv,D} + \lambda_4 \, \mathcal{L}_{Adv,D_{Roi}} \end{aligned} \tag{6.10}$$

with $\lambda_1 = 10$, $\lambda_2 = 75$, $\lambda_3 = 1$, $\lambda_4 = 100$. Additionally, we weighted the shape reconstruction loss $\mathcal{L}_{S-rec}$ more heavily at pixels close to the center of the modification by multiplying the error $\mathcal{L}_{S-rec}$ at each pixel with a 128×128 pixel weight map W. This weight map W was calculated by projecting the Euclidean distance of each vertex in 3D between S^n and S^n_{mod} and subsequent normalization between zero and one. Using this weight map W, we aimed to both increase convergence speed and facilitate the prediction task by weighting shape reconstruction errors more lightly at regions of the neck, the forehead, and the ear. For optimization we used Adam with $\beta_1 = 0.5$, $\beta_2 = 0.999$ and a batch-size of 16. We trained G, D, and D_{Roi} with a learning rate of $lr = 10^{-5}$ over 1 million iterations. For initial experimental runs, we experienced heavy difficulties in stabilizing the training, and in general, we observed slow convergence speed and poor image quality of the modifications. To achieve a better initialization, we pre-trained our model over another 1 million iterations using supervised learning on a synthetic dataset, which comprised corresponding ground-truth images I^n_{mod}. To create this synthetic dataset, we used the OpenGL library to render random faces using the BFM2009 face model [140], random expressions that we randomly drew from the 300W-LP dataset, and random background images from the indoor dataset by Quattoni et al. [151]. Example images of our synthetic pretraining can be seen in Figure 6.4. This dataset enabled supervised learning on synthetic face templates to pre-train our model. For this, we optimized $\mathcal{L}_G$ in (6.10) except that we replaced $\mathcal{L}_{I-rec}$ with a synthetic loss $\mathcal{L}_{Syn}$:

$$\mathcal{L}_{Syn} = \mathbb{E}_{I^n_{mod}, \tilde{I}^n_{mod}} \left[\mathcal{L}_1 \left(I^n_{mod}, \tilde{I}^n_{mod} \right) \right] \tag{6.11}$$

After pre-training our model on synthetic data, we trained our model 3DDFA dataset by linearly increasing the region-of-interest-size of D_{Roi} from 16×16 pixels to 48×48 pixels over the first 500 000 iterations. For the remaining 500 000 iterations, we used a random uniform region-of-interest-size between 32×32 pixels and 48×48 pixels. In total, we trained our model for 1 million iterations on the synthetic data and another 1 million iterations on the 300W-LP dataset, which took approximately fifteen days on an NVIDIA RTX2080Ti.

Network architectures. The detailed architectures of our models are given in Appendix B.2. For the generator G, we used the tiramisu U-Net [152]. G received six channels with an image resolution of 128×128 pixels as input which comprised the unmodified image I^n as well as the PNCC of the modified 3D shape S^n_{mod}. The output of G comprised three RGB channels for the predicted modified image $\tilde{I}^n_{mod}$. For the discriminators, we used PatchGAN [115] architectures. For the shape estimator G_S, we used another tiramisu U-Net which received I^n as input and predicted the projected PNCC of S^n. As described in Section 6.3.2, G_S predicted both a mask and a color map of the PNCC. Therefore, the output of G_S comprised five output channels: two channels for the mask (background and face pixels) and three channels for the color values of the PNCC.

6.4 Experiments and Results

We conducted three experiments to evaluate the performance of our model G on both "in-the-wild" images and on a clinical example. Hereby, we evaluated G qualitatively on selected samples of the AFLW2000 dataset in experiment 6.4.1 and quantitatively

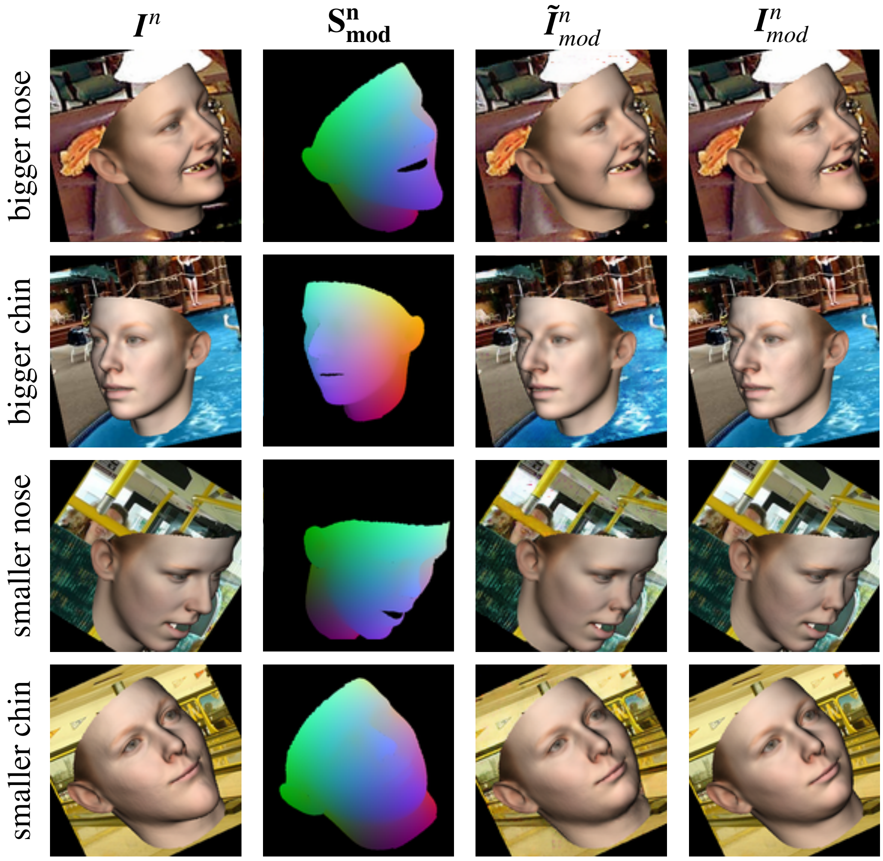
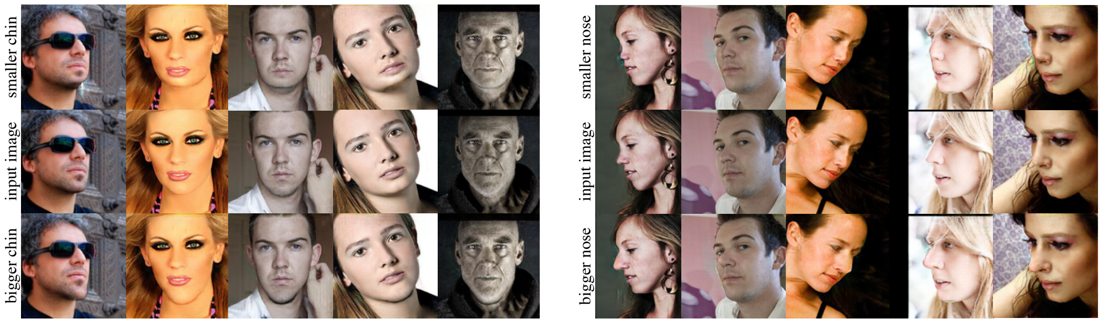


Figure 6.4: Four randomly chosen images of the synthetic dataset used for pre-training G. Each row shows one of the four image modifications that were considered in this study. The figure shows synthetically generated samples of the training set after training G over 1 million iterations. The columns show the input image I^n, the modified input shape S^n_{mod}, the prediction $\tilde{I}^n_{mod}$ of G and the synthetic ground-truth I^n_{mod}.

in experiment 6.4.2. Lastly, we aimed to predict the post-operative face using G in experiment 6.4.3.

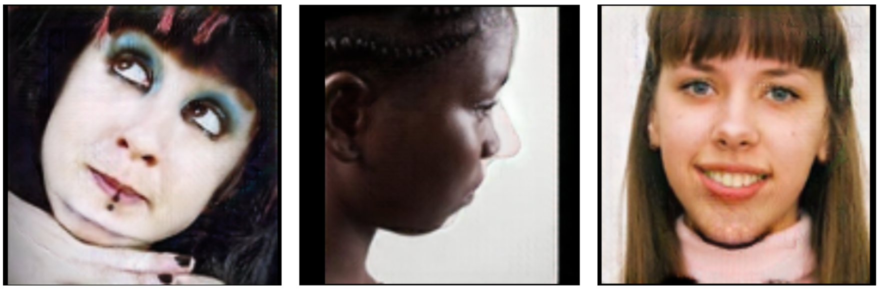
6.4.1 Qualitative Results

Experiment. We evaluated G on the AFLW2000 dataset [126], which yields 2000 pairs of images and corresponding shape parameters of the statistical point distribution model as well as the camera parameters to project the 3D shape to the 2D image plane. Like the 300W-LP dataset, these 3D faces were estimated using a semi-automatic fitting procedure [140] and were assumed as ground-truth in this study. For inference, we tested our model on all 2000 images of the AFLW2000 dataset using all four different modifications S^n_{mod} as input that was proposed in this study and are visualized in Figure 6.2 (b): larger chin, smaller chin, larger nose, and smaller nose.



(a) (b)

Figure 6.5: Selected predictions on the AFLW2000 dataset. The middle row shows the original images I^n which were used as input for G. The top rows show predictions of a smaller chin (a) and nose (b), respectively. The bottom rows show predictions of a larger chin (a) and nose (b), respectively.



(a) (b) (c)

Figure 6.6: Three most frequent types of “failure” that we observed for predictions on the AFLW2000 dataset. In (a) our model was tasked to predict a smaller chin. However, the original chin is still visible which results in an unrealistic prediction of the background. (b) shows the prediction of a nose enlargement where the model only generated the outlines of the desired shape of the nose. In (c) the enlarged chin of the woman yields an unnatural dark texture which was frequently observed for chin enlargements of female faces.

Results. Figure 6.5 (a) and Figure 6.5 (b) show the predictions of G for selected images that we visually judged to be both realistic and accurate compared with the given input S^n_{mod}. In detail, the top rows show the predictions $\tilde{I}^n_{mod}$ for a smaller chin and a smaller nose, respectively, while the bottom row shows the predictions for a larger chin and nose. For comparison, the original unmodified images I^n are given in the middle row. As can be seen in our best examples, G was able to modify the desired region for images with varying head pose and illumination settings. The applied modification appeared to be realistic, and the integration of the modified face with the rest of the face was plausible. Notably, our model was also able to predict a plausible background of regions that were previously occluded by face, as seen in the top row of the figures. However, the overall performance was moderate as our model did not consistently predict realistic and accurate facial modifications on all images of the dataset. As an example, Figure 6.6 shows three of the most frequent types of “failure” that we observed on the AFLW2000 dataset. In Figure 6.6 (a), G was tasked to predict a smaller chin. However, the model did not remove the previously larger part of the chin, which resulted in an unrealistic prediction of the background. Figure 6.6 (b) shows the prediction of a nose enlargement. However, the model only generated the outlines of the desired shape of the nose, which was sufficient to fool the shape estimator G_S and achieve a low shape reconstruction error $\mathcal{L}_{S-rec}$. These cases of failure could be found for both chin enlargements and nose enlargements and suggest a weak adversarial loss $\mathcal{L}_{Adv}$. Lastly, we observed a specific case of failure that mostly affected predictions of large chins in women. As seen in the example in Figure 6.6 (c), the enlarged chin of the female face yielded an unnatural dark texture at the tip of the chin, which could be interpreted as either artifact, facial hair, or heavy shading and generally resulted in chin predictions that appeared more manly compared to the overall appearance of the original face. A potential explanation for this observation is given in Section 6.5. The described cases can also be seen in Figure 6.7, in which we show the predictions for all four modifications on the first eight samples of the AFLW2000 dataset to provide the reader with an unbiased selection of images.

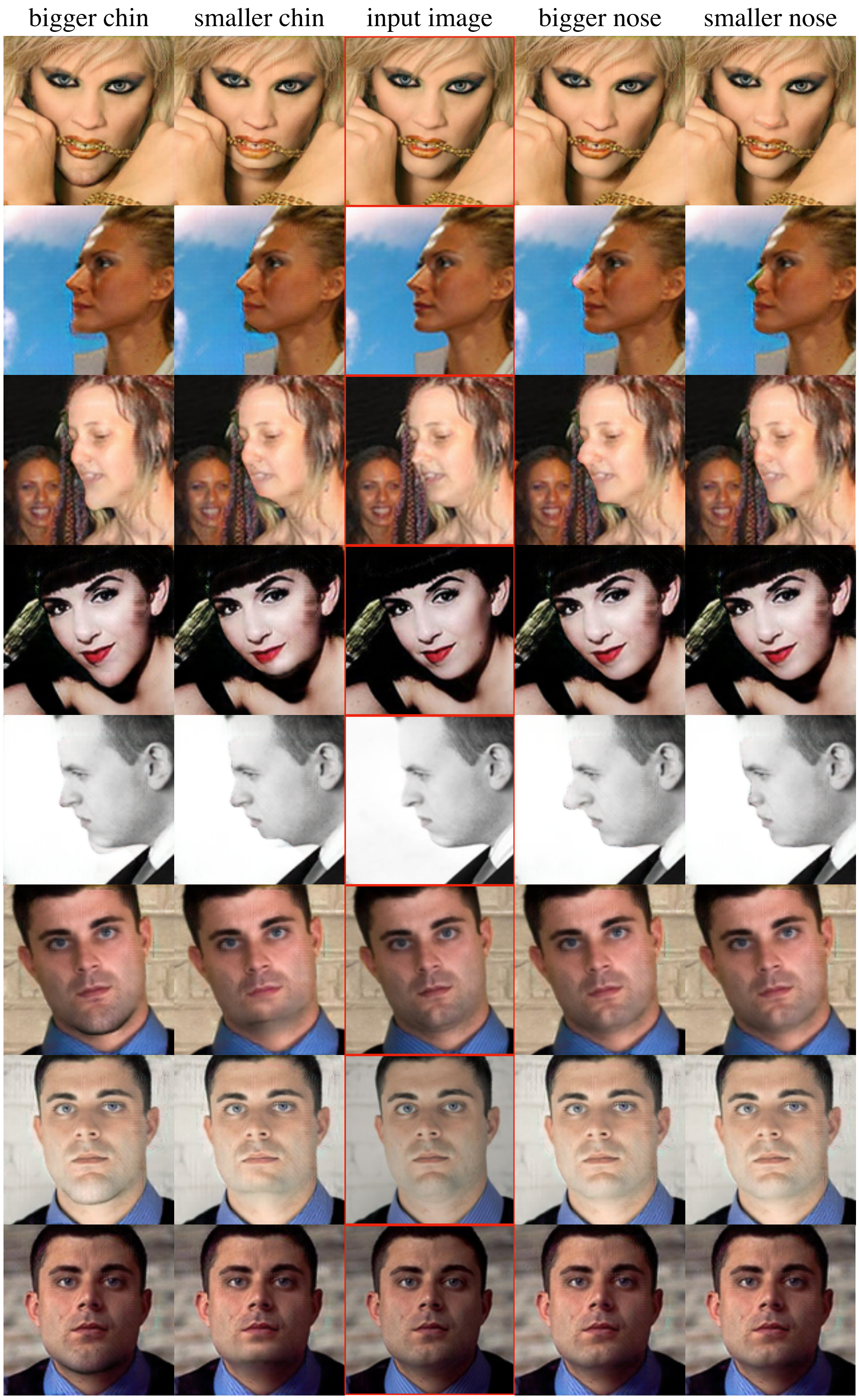


Figure 6.7: Nose and chin modifications on the first eight AFLW2000 dataset samples. Third column shows the baseline input images I^n for the generator G. First and second column show the predictions of G for a bigger chin and a smaller chin, respectively. Fourth and fifth column show the predictions of G for a bigger nose and a smaller nose, respectively.

6.4.2 Quantitative Results

Experiment. In this section, we aim to analyze the accuracy of the predicted modified regions in $\tilde{I}^n_{mod}$. To enable a fully automatic approach, we used a facial landmark predictor [153] and calculated the normalized Euclidean distance between landmarks of $\tilde{I}^n_{mod}$ and the projected landmarks of the corresponding shape S^n_{mod}.

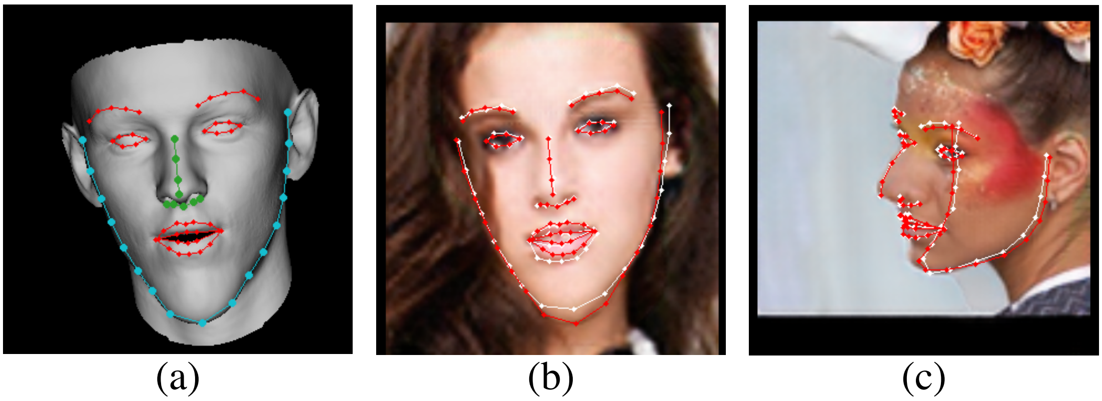


Figure 6.8: Facial landmark annotations to evaluate the accuracy of our model. (a) shows the 68 facial landmarks that we annotated on the 3D shape via the vertex indices. Landmarks of the chin region #[1-17] are annotated in cyan. Landmarks of the nose region #[28-36] are annotated in green. (b) shows the predicted landmarks in white on the prediction $\tilde{I}^n_{mod}$ for an enlarged chin. The ground-truth landmarks derived from the modified shape S^n_{mod} are shown in red. (c) shows the landmark annotations on the prediction $\tilde{I}^n_{mod}$ for an enlarged nose.

To generate $\tilde{I}^n_{mod}$, we rerun the pipeline on all 2000 images of the AFLW2000 as described in section 6.4.1. To calculate the "ground-truth" facial landmarks, we annotated 68 landmarks on the 3D shape S^n_{mod} via their indices provided by [154]. After that, we projected the landmarks in 3D to the 2D image plane as visualized in Figure 6.8 (a). Additionally, we annotated the first 17 landmarks (#[1-17]) to belong to the chin region and nine landmarks (#[28-36]) to belong to the nose region as annotated in Figure 6.8 (a). We then predicted all 68 facial landmarks of the modified faces $\tilde{I}^n_{mod}$ using the face-alignment network by Bulat et al. [153]. For comparison, we also predicted the facial landmarks of the original images I^n and calculated the ground-truth facial landmarks using S^n. Examples of the predicted 68 landmarks and the corresponding ground-truth landmarks are given in Figure 6.8 (b), (c). To create a comparable setting to [153], we also up-scaled all images from 128×128 pixel to 450×450 pixel resolution before applying the landmark predictor and calculating the normalized mean error (NME) proposed by [153]:

$$\text{NME} = \frac{1}{N}\sum_{k=1}^{N}\frac{\|x_k - y_k\|_2}{d} \tag{6.12}$$

Hereby, x was the landmark predictions, y was the projected landmarks of the 3D shape, and $d = \sqrt{w \times h}$ was a normalization factor derived by the width w and height h of the bounding boxes given by the AFLW2000 dataset for each unmodified image I^n. Additionally, we separately calculated the NME for the chin region using the landmarks #[1-17] with $N = 17$ and for the nose region using the landmarks #[28-36] with $N = 9$ as shown in Figure 6.8 (a).

Results. Figure **??** shows the cumulative distribution functions (CDFs) of the NME across all samples of the AFLW2000. In both figures, a baseline CDF is provided as a dotted line in red which was calculated on the original baseline images I^n using all 68 landmarks #[1-68]. When compared to the reported results in [153], we were able to reproduce similar baseline CDFs, and thus, we are confident that we correctly implemented the face-alignment framework and the NME calculation described by Bulat et al. [153]. For a meaningful analysis of the prediction accuracy of G, the CDFs of the modified images $\tilde{I}^n_{mod}$ must not be interpreted on their own since prediction errors of G might be confused with prediction errors of the landmark predictor or fitting errors of the 3D shapes in the dataset. Instead, we compared the CDFs of the baseline images I^n with the CDFs of the modified images $\tilde{I}^n_{mod}$ in an attempt to compensate the landmark prediction errors and the fitting errors of the dataset. Figure **??** (a) shows a CDF in cyan solid line which was calculated using the landmarks of the chin #[1-17] on the unmodified baseline images I^n. The CDFs of the modified images $\tilde{I}^n_{mod}$ are given for a larger chin as a black dashed line and a smaller chin as a gray dash-dotted line. Likewise, Figure **??** (b) shows the CDFs using the landmarks of the nose #[28-36] on the baseline images I^n as a solid green line and the predictions $\tilde{I}^n_{mod}$ of the modifications (larger nose as a black dashed line, smaller nose as a gray dash-dotted line). For a better visual comparison, we cropped both figures on the x-axis at 13.25% and 3.68% to exclude the worst 3% of all calculated NME errors that belonged to the baseline CDFs #[1-17] shown as a cyan solid line and #[28-36] as a solid green line, respectively. As reported by [153], these high NMEs that we excluded were mostly attributed to poor ground-truth annotations of the AFLW2000 dataset or faces in the background that led to wrong landmark predictions. As seen in Figure **??** (a), the CDFs for larger or smaller chins were comparable to the CDFs of the baseline #[1-17]. Quantitatively, the normalized area under the curve (AUC) of the baseline #[1-17] was slightly worse with an AUC of 57.13% compared to the larger chin predictions with an AUC of 59.11% and smaller chin predictions with an AUC of 57.43%. For the nose modifications, the CDFs were worse compared to the baseline #[28-36] which led to a baseline AUC of 51.39%, a larger nose AUC of 47.62%, and a smaller nose AUC of 45.58%. Thus, our quantitative results on our in-the-wild dataset suggested that our model predictions were more accurate for chin modifications compared to nose modifications. These quantitative findings are in accordance with our qualitative findings, where we visually observed that the predictions of the chin modifications appeared to be both more realistic and more accurate compared to the modifications of the nose.

The AFLW2000 dataset yielded a high variation of head pose rotations around the vertical axis with the angle θ, which might have been an additional challenge to G. To analyze the effect of such head rotations around θ, we provide the AUCs for three different absolute ranges of the vertical axis in Table 6.1. Hereby, we calculated the AUCs on all baseline images and modified images using trapeze integration and using the same boundaries for the x-axis that are given in Figure **??** (a) and **??** (b), respectively. Additionally, we also normalized each AUC by dividing by the respective length of the x-axis. When comparing the baseline AUCs in Table 6.1 for different angles, one can see that the AUCs strongly decrease for larger θ which can be attributed to a worse prediction accuracy of the landmark predictor as previously reported by [153]. To allow a better comparison with the baseline, Table 6.1 shows the difference between the AUC of the modified images $\tilde{I}^n_{mod}$ and the AUC of the baseline images I^n for each modification and angle θ. For the chin modifications, the AUCs show no clear indication that large head rotations impede the prediction accuracy of G when compared to the baseline AUCs. While the AUC for large angles ($\theta \geq 60°$) decreased for larger chins by 2.04%, the AUC for smaller chins even improved by 7.95% compared to the baseline. This improvement of the AUC compared to the baseline might be explained by a better alignment of the prediction $\tilde{I}^n_{mod}$ with S^n_{mod} compared to the given sample pairs I^n and S^n of the AFLW2000 dataset. For the nose modification, in contrast, larger head rotations with $30° \leq \theta < 60°$ resulted in a strong decrease of the AUC by 11.81% and 6.52% with $60° \leq \theta \leq 90°$. On the other hand, the AUCs for the predictions of smaller noses showed no clear tendency for large θ. Hereby, one should keep in mind that for a frontal view of the head with $\theta = 0°$, the landmark predictions are highly insensitive against modifications of the nose length. Therefore, inaccurate predictions of the nose by G might still yield low NMEs for small θ. As a consequence, the AUCs for small head rotations $\theta < 30°$ should be interpreted with care when regarding the nose modifications. However, overall, the lower AUCs for the nose modifications with $\theta \geq 30°$ suggest that the accuracy of the "larger nose" predictions was poor compared to the baseline, while the landmark accuracy of the "smaller nose" predictions was comparable to the baseline.

Table 6.1: Area Under the Curve of the Baseline and the Modified Images

head pose θ (°)	number of images	baseline #[1-17] AUC (%)	larger chin AUC diff. (%)	smaller chin AUC diff. (%)	baseline #[28-36] AUC (%)	larger nose AUC diff. (%)	smaller nose AUC diff. (%)
[0 - 30)	1312	61.16	+1.06	-1.08	54.76	-4.62	-5.02
[30 - 60)	383	57.60	+0.73	+3.86	51.56	-11.81	-2.84
[60 - 90]	305	45.11	-2.04	+7.95	40.68	-6.52	-0.96

6.4.3 Prediction of the Post-Operative Face

Experiment. In this section, we aim to demonstrate the potential of our approach to predict 2D images of the post-operative face. For this, we evaluated our model on pre-operative and post-operative measurements of four patients that underwent orthognathic surgery to serve as a proof-of-concept. The images of the patients before surgery are given in Figure 6.9 (b) for a frontal and lateral head position, respectively. Before surgery, patients P1, P2, and P3 suffered from a class III malocclusion, and P4 suffered from a class II malocclusion. All four patients were treated by bimaxillary surgery, and the resulting post-operative faces can be seen in Figure 6.9 (d) which were captured eight weeks after surgery. To test our model for the prediction of post-operative outcome, we passed two inputs to our model G: A cropped image of the pre-operative face as seen in Figure 6.9 (b) and a simulation of the post-operative 3D shape that we derived from the surgical planning tool IPS CaseDesigner® [118] as seen in Figure 6.9 (a). More precisely, we used a CT image scan from the pre-operative face, segmented soft-tissue, and bone tissue, and subsequently applied a bimaxillary surgery to the virtual bone structure of the jaw. We used IPS CaseDesigner® to simulate the soft-tissue deformations induced by the correction of the underlying bone structure. Based on this prediction of the post-operative 3D shape, we iteratively fitted a surface template of the BFM2009 model [140] on the 3D virtual face by adopting the approach described in [155]. Next, we estimated the camera matrix to project the fitted surface template onto the pre-operative face in Figure 6.9 (b) by aligning the surface mesh to the upper half of the pre-operative face. The resulting projection of the simulated post-operative face to the image plane can be seen in Figure 6.9 (a). We converted the projected surface mesh to a PNCC by encoding the color of the surface template via their vertex indices. Lastly, we passed the resulting PNCC and the pre-operative image to our model G to predict a 128×128 pixel image of the post-operative face shown in Figure 6.9 (c). Note that we only trained G on the 3DDFA dataset, i.e., the model was never shown images or shape modifications from our clinical test case.

To compare our predictions in Figure 6.9 (c) with the ground-truth post-operative face in Figure 6.9 (d), we aimed to calculate the distance between two images of a face which we call face distance in the following. Hereby, we used a face-recognition neural network to mitigate the effect of changing illumination, skin color, hair color, hair cut, and other changes that were not related to bimaxillary surgery when comparing the prediction of the post-operative face with the post-operative ground-truth. More precisely, we calculated an embedding of each image using the InceptionResnetV1 by Esler et al. [156] which was trained on the VGGFace2 dataset for face recognition [157]. Then, we measured the Frobenius norm between two embeddings to calculate the distance between two images of a face. For every row in Figure 6.9, we measured the face distance between the model's prediction of the post-operative face in Figure 6.9 (c) and the ground-truth post-operative face in Figure 6.9 (d). For reference, we also calculated the face distance between the pre-operative face in Figure 6.9 (b) and the post-operative face Figure 6.9 (d). Next, we also calculated the structural similarity index (SSIM) [158] between the prediction and the post-operative image. Hereby, we first manually aligned the predicted image with the post-operative face, converted the image to a grayscale, and performed a histogram matching with the post-operative face to mitigate differences in illumination and skin color. Again, we also calculated the SSIM between the pre-operative face and the post-operative face to serve as a reference. Lastly, we showed our predictions to two clinician experts with years of experience regarding cranio-maxillofacial surgery to form a combined statement about the context of this study as well as the perceived quality of the predictions.

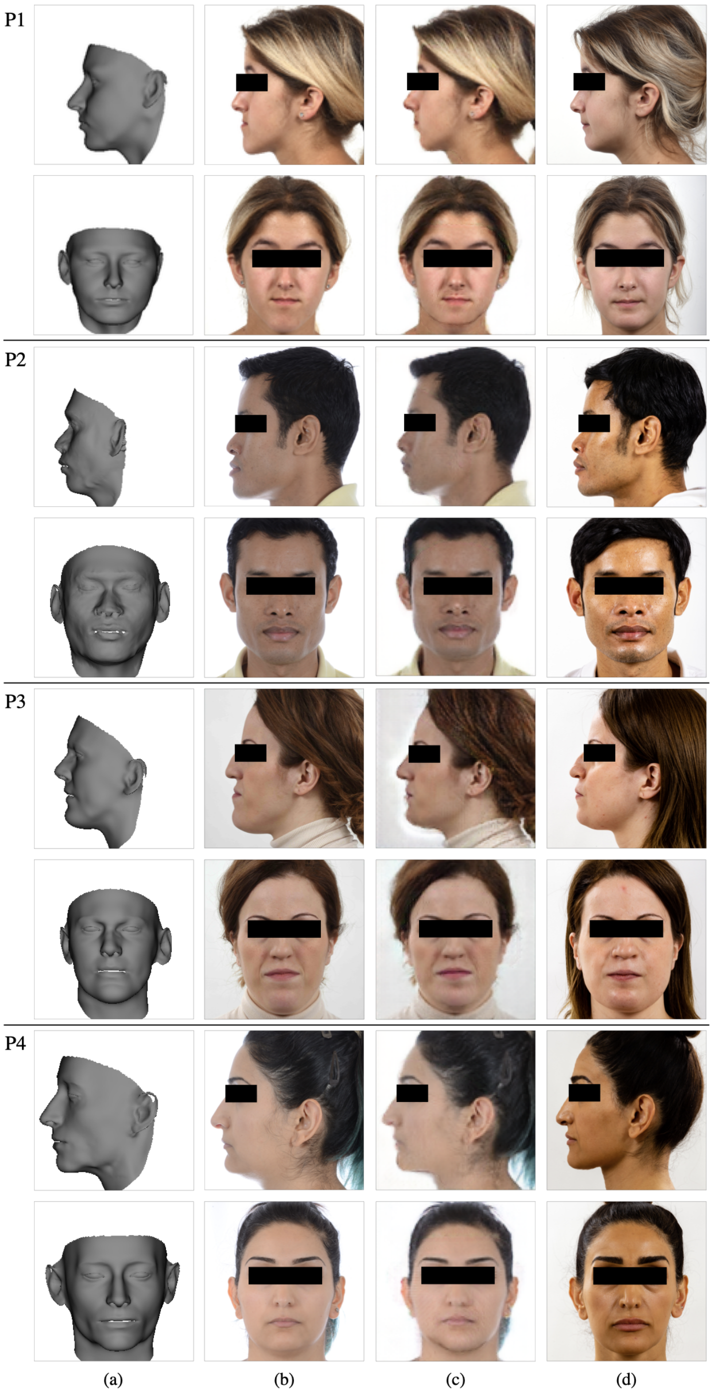


Figure 6.9: Prediction of the post-operative face on four clinical examples. The upper row shows each patient's inputs, predictions, and ground-truths in lateral view. The lower row shows the same patient in a frontal view. (a) shows the fitted and projected face templates derived from a simulation of a surgery planning tool to predict the 3D shape of the post-operative face. The visualized 3D shapes were converted to PNCCs and passed as input to our model. (b) shows the images of the patient's face before cranio-maxillofacial surgery, which were passed as a second input to our model. (c) shows the predictions of the post-operative face generated by our model. For a better visibility, the images were up-scaled from the original output resolution of 128×128 pixels. (d) shows the patient's face eight weeks after cranio-maxillofacial surgery as a ground-truth. All images were published with the patient's consent.

Results. The resulting predictions $\tilde{I}^n_{mod}$ of the post-operative face are shown in Figure 6.9 (c) for the frontal and the lateral view. Additionally, the face distance and the SSIM are given in Table 6.2. Hereby, two identical faces would ideally lead to a face distance of 0, and two identical images would lead to an SSIM of 1. 4 and 0 correspond to the opposite, respectively.

When comparing the lateral view of the pre-operative face in Figure 6.9 (b) with the predictions for the lateral view, one can see a clear upward and right shift of the chin for all three patients P1, P2, and P3. For the lateral view of P4, the difference between the pre-operative face and the prediction was less clear. Note hereby that P4 suffered from a different class of malocclusion which made the difference between the pre-operative face and the post-operative face visually less obvious. Nonetheless, the face distance in Table 6.2 was smaller for all lateral predictions, which indicates a closer resemblance to the ground-truth image of the prediction compared to the pre-operative face. For the frontal views, the prediction of P1 yielded a clear upwards shift, while the effect of cranio-maxillofacial surgery on the frontal prediction of P2 and P3 was less pronounced. For P4, however, the frontal prediction of the chin appeared unnatural and strongly differed from the ground-truth in Figure 6.9 (d). Accordingly, the face distance between the prediction and the ground-truth in Table 6.2 increased compared to the face distance between the pre-operative face and the ground-truth. Thus, the face distance measurements suggest the frontal prediction of P4 to be a failed example of our approach. In contrast, the SSIM scores between the predictions and the ground-truth were slightly higher compared to the reference for all patients except for the frontal and lateral predictions of P2. This suggests that the prediction of the post-operative face of P2 was less accurate. On the other hand, the differences between the SSIM scores of the prediction and the reference were only minor, i.e., less than 0.2 %.

Overall, the facial appearance of a majority of the predictions in Figure 6.9 (c) were similar to the facial appearance of the ground-truth images of the post-operative face in Figure 6.9 (d). Notably, our model also predicted a white background on the lateral views in Figure 6.9 (c) and plausible backgrounds of the throat on the frontal views in Figure 6.9 (c). On the other hand, minor artifacts were present in all predictions, which included locally blurred regions of the face (particularly at the lips) and smaller artifacts on the skin, the lips, and the throat region. Hereby, one should keep in mind that model G was never trained on modifications S_{mod} that substantially altered regions of the mouth or the throat. Additionally, when comparing the silhouette of the pre-operative face of P1 in the top row in Figure 6.9 (b) with the simulated post-operative face in Figure 6.9 (a), one can see a shape difference at the throat which had to be adapted by G. A particular reason for this deviation of the simulated shape compared to the pre-operative face in Figure 6.9 (b) might be that the simulated post-operative face was derived from a CT scan which means that the patient was lying on the CT table. Therefore, the head position and the tissue of the throat might have varied compared to the up-right position of the patient during the capturing of the image in Figure 6.9 (b). Conveniently, our model

Table 6.2: Face Distance and SSIM between the Post-Operative Image (Ground-truth) and the Pre-Operative Face or the Post-Operative Prediction, respectively.

patient	view	face distance to the post-operative image		SSIM to the post-operative image	
		pre-operative	prediction	pre-operative	prediction
P1	lateral	0.924	0.894	0.9847	0.9851
	frontal	0.758	0.632	0.9735	0.9750
P2	lateral	0.718	0.656	0.9853	0.9838
	frontal	0.398	0.398	0.9708	0.9705
P3	lateral	1.092	1.051	0.9893	0.9907
	frontal	0.683	0.676	0.9112	0.9112
P4	lateral	0.647	0.639	0.9866	0.9867
	frontal	0.747	0.819	0.9823	0.9824

learned to almost fully ignore shape variations of the neck since a) the ground-truth shapes of the neck were poorly aligned with the images in our 3DDFA training set and b) the shape reconstruction loss $\mathcal{L}_{S-rec}$ for necks was only lightly weighted as described in Section 6.3.4.

Lastly, we provide the statement of our clinical experts on the context of this study and the perceived image quality of the predictions: "In most cases, patients want to see realistic photographs of their facial appearance after correction of a skeletal deformity. Commercial software offers a 3D mesh with the possibility of texture overlay, still most patients are not able to identify with the displayed data. The provided predictions appear realistic and arguably closer to a natural image of a face a patient can relate to. As there is a risk of body dysmorphophobic disorder in severe changes to the facial appearance, preparing the patient with a relateable prediction and adequate counseling before obtaining informed consent for the procedure. The contours of the predicted faces appear smooth while the predictions of our commercial software produces uneven contours at the jaw after simulating the planned surgery protocol. [Authors note: These uneven contours can be seen at the jaw in Figure 6.9 (a). on the lateral view of, e.g., subject P2 simulated using [118] or, for instance, in Figure 6 of [123].] On the other hand, the applied face modifications partly appear a bit extreme (compare, e.g., the lateral prediction of P1 and P2), and the contours of some predictions are locally ambiguous, in particular at the lips. The main advantage we see, however, is that the proposed approach does not require a 3D texture scanner. While every clinical site has a CT scanner and a camera, the availability of compatible 3D camera systems remains limited. In such cases, patients would have to make their decision based on a prediction that looks similar to Figure 6.9 (a), which is inadequate. Using the approach proposed in this study, however, we might be able to provide the patient in the future with a fast, non-committal, and natural-looking prediction of her/his face after only one CT scan."

6.5 Discussion

In the results in Section 6.4.1, we showed that our CycleGAN was capable of predicting realistic and recognizable modifications of the chin and the nose on selected examples. Subsequently, we aimed to measure the accuracy of our predictions in Section 6.4.2 by evaluating the Euclidean distance of facial landmarks on the AFLW2000 dataset. Hereby, we found the accuracy of the chin modifications to be similar compared to the accuracy of the matched 3D shapes of the AFLW2000 dataset. For the nose modifications, we found worse accuracy across the dataset compared to the baseline which was particularly pronounced for large head rotations. Lastly, we showed our proof-of-concept on four clinical patients where we predicted 2D images of the post-operative face according to the soft-tissue deformations of a surgery planning tool. Through this, we demonstrated that our model was indeed able to apply realistic modifications on four clinical patients without requiring additional training. Concerning the desired use case in clinical practice, one could argue that the task to train on "in-the-wild" images was a much more difficult task (in particular background prediction, illumination, and head pose) compared to the expected more controlled environment of the clinical use case. Thus, the results of our model might improve for a training dataset that is closer to the clinical test case. Likewise, one could also argue that the training on modifications of the nose was not required to predict the post-operative face in Section 6.4.3, which only affected the jaw. Therefore, we like to note that cranio-maxillofacial surgeries, in general, are not only concerned with jaw deformations but other regions of the face as well, and, in particular, nose modifications can have a strong impact on the identity or appearance of the patient's face. Thus, our motivation was to propose a more generalized approach that theoretically enables modifications of any facial region that can be represented by both the statistical shape model and the dataset. Such an approach would require a model G that learned a continuous understanding of the desired 3D shape and accordingly applied facial modifications wherever the 3D shape differed from the given input image. However, we found qualitatively and quantitatively that our model performed worse on nose modifications than on chin modifications in terms of robustness and accuracy. To explain the worse performance of our model on nose manipulations, we suggest the following reasons: First, we hypothesize that modifying noses is a much more complex task to solve compared to chins as noses yield arguably more fine-detailed textures and vary more strongly across different head poses. Consequently, this would suggest that our proposed approach to predict the post-operative outcome of faces might be limited at the moment to spatially less complex structures like the chin. On the other hand, our training procedure might still be biased in favor of chins. Although the number of chin and nose modifications was balanced, and we used a weight map in the reconstruction loss to account for the size differences between noses and chins, our discriminator might have been more sensitive to detect unrealistic chins due to the larger affected area in the image.

From a theoretical point of view, the suitability of our proposed training strategy for manipulating faces using CycleGANs can be discussed controversially. While our method has the advantage of not requiring ground-truth images I^n_{mod} or knowledge of physical models, one could argue that a training strategy based on GANs will always bias the predicted face towards the mean face to maximize the expected reward from the discriminator. Therefore, attempts to manipulate facial properties that are "far away" from the mean face, like an extremely enlarged chin, might result in closer predictions to the mean face, i.e., less "extreme" than the desired modification. Similarly, the model might have learned that extremely enlarged chins are far more likely to belong to male faces. This hypothesis might explain why we found some predictions for enlarged chins of female faces to yield facial hair or artifacts, resulting in a more manly appearance of the chin in Figure 6.6 (c). Additionally, our approach using neural networks might also be vulnerable to ethnic imbalances of the training dataset. Consequently, applying our trained model to predict post-operative outcome might end up favoring, e.g., patients of light skin color by providing a higher prediction quality compared to faces of dark skin color. However, such a racial bias is unacceptable for a clinical use case and would have to be ruled out by thorough testing before considering a model for a clinical application.

The above passages highlighted both empirical and theoretical limitations of our current model to modify facial regions and predict the post-operative face. However, despite these current limitations, we are confident that these challenges can be overcome in the near future, especially in light of the rapid advances of GANs in recent years. In more detail, we particularly would like to improve the training and regularization strategy of the adversarials, the accuracy of the shape estimator G_S, and the image quality, accuracy, and ethnic balance of the dataset used for training. Afterwards, we would like to test our approach for the prediction of medical outcomes in a thorough clinical study. Hypothetically, one might go even further and replace our current representation of the 3D soft-tissue with a 3D bone structure of the jaw. To achieve this, one would have to train a model to estimate the jaw's bone structure from 2D images directly and subsequently train a CycleGAN to manipulate 2D images based on a modification of the bone structure provided by the physician. Having such a model, physicians would have a fast and cheap means to directly predict the post-operative face from 2D images without the need for expensive and time-consuming tomography scans.

6.6 Conclusion

In this study, we introduced a novel idea to predict the post-operative face using a neural network. Hereby, we showed that our prototype model was indeed capable of generating realistic predictions of the patient's face after cranio-maxillofacial surgery according to a given soft-tissue simulation. To train our model, we proposed a novel CycleGAN strategy to learn to modify facial regions of "in-the-wild" images according to a 3D plan of facial shape. Compared to current approaches to render the post-operative face, our approach can directly translate and manipulate the facial texture of a patient in 2D and therefore does not require the acquisition of 3D texture scans. Moreover, we achieved this prediction by merely training our model on open-source images without requiring clinically relevant face modifications or hand-crafted physical models. Based on our preliminary results and the rapid improvements of GANs in recent years, we believe that our proposed approach has a high potential to help the patient in their decision process in favor or against surgery. In future work, we aim to increase the robustness of our model and test our model to predict the post-operative face in a clinical follow-up study.

PART III

FINAL REMARKS

Chapter 7

Conclusion

The main objective of this thesis was to counteract the lack of sufficient images and videos in medical datasets for the training of neural networks (NNs) and thereby contribute novel approaches to the state-of-the-art in this field. Two different approaches, a model-based, described in Section 4.3.2, and generative adversarial network (GAN)-based, described in Section 6.3, were developed, and their performance was investigated by training an NN using a dataset of three different surgical procedures.

The study of Al Hajj et al. [15] and Twinanda et al. [21] showed that it is beneficial for the performance to include the temporal component. The temporal component contains coded information about the operation process, and the occurrence of surgical instruments generally correlates with the progress of the surgical intervention. However, the number of recorded videos is not sufficient for the end-to-end training of an NN with temporal modeling. For this purpose, the model-based approach, also called workflow augmentation, was developed to generated new artificial videos using the workflow of events and the individual event video segments. Both were retrospectively extracted from the existing dataset. With the rule-based recombination of events, the variation of the event combination in the training's dataset can be increased, and thus previously non-existent combinations of events can be generated. This was demonstrated in the *Workflow Augmentation of Video Data for Event Recognition with Time-Sensitive Neural Networks* described in Chapter 4 using cataract surgery. The surgical tool classification in surgical videos was improved with the help of the novel approach. Thereby, the original dataset was enlarged and enhanced using the developed workflow augmentation method, i.e., the variability in the videos was increased, and the distribution of the video length was balanced. The augmentation resulted in an improved classification accuracy (ACC) of the 33 surgical classes of surgical tools and tool combinations by 2.8 % to 93.5 %.

Since the results were promising, it was consequential to investigate in a second step if the semantic information in the dataset that was used to generate the new artificial videos can be augmented by the workflow augmentation method. The hypothesis was that augmentation enhanced the recognition performance of the semantic information. For this purpose, the project with the title *Improving Surgical Phase Recognition in Videos using*

Workflow Augmentation, described in Chapter 5 was conducted. Thereby the example of laparoscopic cholecystectomy surgery is used to investigate: how the recognition performance of semantic information, i.e., surgical phases, can be improved with the workflow augmentation method compared to the literature. Thereby, the challenge is that the desired information is encoded in consecutive frames rather than individual frames. Using the workflow method to augment the training dataset, an improvement in phase recognition of 8.7 % to an ACC of 96.96 % was achieved compared to other approaches in the literature.

In the two examples, the developed workflow augmentation approach could mitigate the lack of adequate data very well, but some limitations cannot be compensated by this approach and should be mentioned. The performance of the NN is most likely not sufficient for the application in the medical field. An ACC of 93.5 % in the tool and 96.96 % in phase recognition might have harmful consequences for the patient (in six out of 100 and three out of 100 cases). The reason for these inaccuracies in the recognition could be, for example, errors in the annotation of the datasets, as already suggested by Quellec et al. [1]. Such errors exist is generally undisputed and suspected in the presented two examples in this thesis. In the dataset of the cataract surgery, we were able to identify and correct them based on a comparison of the annotations of the number of tools on the surgical tray. However, in the example of cholecystectomy, we could only suspect them in *P4* due to the pattern in the confusion matrix performed in second study. A possible proof could not be performed due to a lack of expertise. Moreover, there are also temporal errors of the annotations in the datasets. In the case of the cataract dataset, the exact time at which a tool contacts the tissue or, in the case of cholecystectomy, the transition between the phases is noise-affected. This can be explained by the definition of the exact time points, which allows more than one possible interpretation, as in the case of the exact phase delimitation, or by technical limitations like frame rate and video resolution in case of the tools which were in tissue contact. However, in the confusion matrices (e.g., Table 5.5, Table A.4) and in Figure 5.9a or Figure 5.9b, we can see exactly this fluttering in predicting the event at the transitions between adjacent classes. Our workflow approach cannot mitigate these inconsistencies. Instead, they are amplified by the redundancy use of the segments, leading to worse performance. Furthermore, surgical workflow events or event combinations that are possible and conceivable in theory (also reported in the literature) may not be included in the dataset because they are very rare or have not yet appeared. At this point, the workflow augmentation method reaches its limits. Although it allows for variations of events in the workflow, on the other hand, it requires real event recordings to integrate them into the artificial videos. Exactly at this point the GAN-based approach, which could generate realistic images of artificial events, could be a remedy.

In the third project, *3D-Guided Face Manipulation of 2D Images for the Prediction of Post-Operative Outcome after Cranio-Maxillofacial Surgery*, described in Chapter 6, we generate a realistic image after cranio-maxillofacial surgery using a GAN, a real patient image, and a modified 3D model. Since there were no suitable datasets that contain pre-

and postoperative images for supervised learning of a NN, a cycle generative adversarial network (CycleGAN) was chosen. A CycleGAN performs an image-to-image translation where the output can act as input with a transformed condition. This approach allows the network to train itself by evaluating the output with an additional quality checker NN and providing feedback to the CycleGAN. However, the available data were still not sufficient for this kind of training. Therefore, in the first step, synthetic portrait photos of 3D computer face models with arbitrary backgrounds were generated, see Figure 6.4. The synthetic data allows a supervised pre-training of the generator of the CyleGAN. However, the synthetic images are not very realistic and vivid, so further training on another dataset was necessary. Since patients were not available in sufficient quantity, the decision was taken to use the 300W-LP dataset by Zhu et al. [126], which contained photographs of celebrities and the parameters for the BFM2009 [140] shape model. Finally, both datasets were used together to train the CycleGAN, that in the end, it was capable of predicting the postoperative images of a patient (see in Figure 6.9).
However, an issue related to the training in the second dataset appeared and is visualized in Figure 6.6(c). The image creates the impression that the woman has a beard. This beard is not present in the original images, which leads to the assumption that the CycleGAN additionally modulated a beard into the image when enlarging the chin. The results are also worse for images of people of color than for other people. A possible explanation for these phenomena could be a biased dataset that is unbalanced and contains more men with beards and people with light skin color. The described problems explicitly demonstrate once again how insufficient data can influence the performance of NN and can cause problems in the generation of synthetic data. Therefore, it is important to have a balanced, error and bias-free input dataset. The meta-information, i.e., features that are not primarily in the foreground and therefore have no label, can also negatively influence the training. Therefore, it is important that sufficient basic diversity, such as gender, skin color, disease characteristics, surgical procedures, etc., is included in the dataset. It is important to keep in mind that the entire image is used for feature extraction, and therefore, the meta-information included unclearly in the decision-making process should also be considered. As a result, gendering or lack of fairness in training can arise.

In conclusion, the methods that were developed in this thesis can partially overcome the lack of samples and datasets, and therefore a better performance of NN could be achieved. The methods can be used to design better artificial intelligence-based medical support systems. The support system can assist the physician in clinical routine, e.g., diagnosis, therapy, or image-guided interventions, reducing the clinical workload and thus improving patient safety.

Chapter 8

Outlook

The thesis explicitly underlines the statement of Lundervold et al. [25] that it is worthwhile to invest effort in developing new methods for the expansion and enhancement of existing datasets. Therefore, it is important to continue on this track in future work. The combination of the workflow augmentation and the generative adversarial network (GAN)-based approaches could considerably increase the performance of neural networks (NNs). With the help of adequate synthetic datasets, better balancing and further unbiasing of datasets could be achieved, or complete synthetic datasets for image and video analysis could be generated according to predefined rules and with the help of computer models. Through verified models, errors in the artificial datasets could be eliminated. Because imbalance and errors in the datasets are the biggest risks in training an NN. Here, it is important to have an automatic method to check the quality of the datasets based on defined parameters and correct them in the best case.

Nevertheless, it is also important to have a sufficient quantity of original data for testing and developing new methods in the future. For this reason, it is important to automate the annotation process of the data as much as possible, although this might initially require more technical or organizational effort. Because a good original dataset is still the best prerequisite for the respective artificial extension of datasets.

Appendix A

Project: Surgical Tool Classification

A.1 Confusion Matrices of different Classifier

Table A.1: Confusion matrix rounded to 4 digits of the workflow augmented test data for the CNN classifier.

Prediction Label	no tool in contact	biomarker	hydrodissection cannula	Rycroft cannula	viscoelastic cannula	cotton	capsulorhexis cystotome	Bonn forceps	capsulorhexis forceps	Troutman forceps	irrigation/aspiration handpiece	phacoemulsifier handpiece	implant injector	primary incision knife	secondary incision knife	micromanipulator	suture needle	Mendez ring	Mendez ring & biomarker	Bonn forceps & secondary incision knife	primary incision knife & Bonn forceps	capsulorhexis cystotome & Bonn forceps	phacoemulsifier handpiece & Bonn forceps	phacoemulsifier handpiece & micromanipulator	irrigation/aspiration handpiece & micromanipulator	hydrodissection cannula & micromanipulator	Troutman forceps & suture needle	Recall
no tool in contact	480524 0.9907	53 0.0001	288 0.0006	289 0.0006	551 0.0011	403 0.0008	268 0.0006	51 0.0001	72 0.0001	190 0.0004	981 0.002	323 0.0007	182 0.0004	46 0.0001	232 0.0005	331 0.0007	25 0.0001	52 0.0001	0	72 0.0001	14 0	22 0	0	50 0.0001	9 0	0	16 0	0.9907
biomarker	272 0.3178	582 0.6799	0	0	1 0.0012	0	0	0	0	0	0	1 0.0012	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6799
hydrodissection cannula	854 0.0431	0	18439 0.9307	2 0.0001	31 0.0016	0	77 0.0039	1 0.0001	21 0.0011	1 0.0001	118 0.006	84 0.0042	5 0.0003	0	0	102 0.0051	0	0	0	1 0.0001	0	5 0.0003	0	57 0.0029	6 0.0003	9 0.0005	0	0.9307
Rycroft cannula	1398 0.1875	0	14 0.0019	5875 0.7879	41 0.0055	9 0.0012	7 0.0009	0	0	0	51 0.0068	5 0.0007	5 0.0007	1 0.0001	6 0.0008	33 0.0044	1 0.0001	0	0	1 0.0001	1 0.0001	1 0.0001	0	2 0.0003	5 0.0007	0	1 0.0001	0.7879
viscoelastic cannula	1546 0.057	0	18 0.0007	11 0.0004	25241 0.9307	0	69 0.0025	5 0.0002	9 0.0003	1 0	97 0.0036	13 0.0005	5 0.0002	7 0.0003	4 0.0001	65 0.0024	0	0	0	7 0.0003	4 0.0001	0	0	9 0.0003	8 0.0003	2 0.0001	0	0.9307
cotton	1402 0.099	0	1 0.0001	3 0.0002	1 0.0001	12746 0.9001	1 0.0001	0	1 0.0001	0	2 0.0001	0	0	1 0.0001	0	2 0.0001	0	0	0	0	0	0	0	1 0.001	0	0	0	0.9001
capsulorhexis cystotome	1667 0.0347	0	102 0.0021	6 0.0001	350 0.0073	0	45463 0.9463	0	74 0.0015	0	164 0.0034	14 0.0003	26 0.0005	7 0.0001	2 0	116 0.0024	0	0	0	3 0.0001	0	23 0.0005	0	12 0.0002	13 0.0003	1 0	1 0	0.9463
Bonn forceps	411 0.1975	0	3 0.0014	0	2 0.001	0	0	1558 0.7487	0	1 0.0005	0	1 0.0005	1 0.0005	0	1 0.0005	1 0.0005	0	0	0	48 0.0231	52 0.025	1 0.0005	0	1 0.0005	0	0	0	0.7487
capsulorhexis forceps	463 0.0349	0	18 0.0014	1 0.0001	24 0.0018	0	47 0.0035	0	12464 0.9408	3 0.0002	159 0.012	18 0.0014	16 0.0012	1 0.0001	0	27 0.002	0	0	0	0	1 0.0001	1 0.0001	0	3 0.0002	1 0.0001	1 0.0001	0	0.9408
Troutman forceps	593 0.0775	0	11 0.0014	0	4 0.0005	1 0.0001	1 0.0001	1 0.0001	4 0.0005	6919 0.9039	48 0.0063	16 0.0021	20 0.0026	0	2 0.0003	20 0.0026	0	1 0.0001	0	0	0	4 0.0005	2 0.0003	5 0.0007	2 0.0003	0	1 0.0001	0.9039
irrigation/aspiration handpiece	2647 0.0156	0	177 0.001	19 0.0001	122 0.0007	0	168 0.001	1 0	59 0.0003	12 0.0001	165619 0.9762	311 0.0018	91 0.0005	9 0.0001	0	168 0.001	0	0	0	5 0	3 0	14 0.0001	5 0	72 0.0004	156 0.0009	3 0	4 0	0.9762
phacoemulsifier handpiece	585 0.025	0	74 0.0032	2 0.0001	6 0.0003	0	10 0.0004	0	5 0.0002	2 0.0001	300 0.0128	22123 0.9458	22 0.0009	1 0	0	16 0.0007	0	0	0	0	0	3 0.0001	19 0.0008	223 0.0095	0	0	1 0	0.9458
implant injector	696 0.0414	0	27 0.0017	5 0.0003	11 0.0007	0	35 0.0021	1 0.0001	5 0.0003	10 0.0006	113 0.0067	38 0.0023	15789 0.94	12 0.0007	3 0.0002	31 0.0018	0	0	0	6 0.0004	7 0.0004	2 0.0001	1 0.0001	2 0.0001	2 0.0001	0	0	0.9401
primary incision knife	448 0.1	0	0	0	20 0.0045	0	8 0.0018	1 0.0002	0	0	15 0.0033	0	20 0.0045	3806 0.8494	21 0.0047	1 0.0002	1 0.0002	0	0	30 0.0067	107 0.0239	1 0.0002	0	0	2 0.0004	0	0	0.8494
secondary incision knife	714 0.1603	0	3 0.0007	2 0.0004	6 0.0013	0	1 0.0002	0	0	0	6 0.0013	0	4 0.0009	11 0.0025	3645 0.8184	1 0.0002	0	0	0	57 0.0128	3 0.0007	0	0	1 0.0002	0	0	0	0.8184
micromanipulator	991 0.0462	0	147 0.0068	13 0.0006	142 0.0066	0	130 0.0061	0	12 0.0006	12 0.0006	242 0.0113	24 0.0011	21 0.001	3 0.0001	0	19692 0.9174	2 0.0001	0	0	0	0	2 0.0001	0	23 0.0011	5 0.0002	3 0.0001	1 0	0.9174
suture needle	215 0.1946	0	3 0.0027	1 0.0009	0	0	1 0.0009	0	0	3 0.0027	11 0.01	0	1 0.0009	0 0	0	3 0.0027	837 0.7575	0	0	0	0	0	0	0	0	0	30 0.0271	0.7575
Mendez ring	321 0.0687	5 0.0011	1 0.0002	0	0	1 0.0002	0	0	0	0	0	0	1 0.0002	1 0.0002	0	0	0	4340 0.9293	0	0	0	0	0	0	0	0	0	0.9293
Mendez ring & biomarker	1 0.0244	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31 0.7561	8 0.1951	0	1 0.0244	0	0	0	0	0	0	0.1951
Bonn forceps & secondary incision knife	200 0.0645	1 0.0003	0	2 0.0006	3 0.001	0	0	28 0.009	0	0	11 0.0035	1 0.0003	6 0.0019	7 0.0023	34 0.011	1 0.0003	0	0	0	2798 0.902	7 0.0023	2 0.0006	0	0	1 0.0003	0	0	0.902
primary incision knife& Bonn forceps	186 0.0327	0	2 0.0004	0	16 0.0028	0	3 0.0005	57 0.01	2 0.0004	1 0.0002	7 0.0012	2 0.0004	15 0.0026	56 0.0098	1 0.0002	0	0	0	0	12 0.0021	5320 0.9356	0	3 0.0005	2 0.0004	1 0.0002	0	0	0.9356
capsulorhexis cystotome & Bonn forceps	158 0.006	0	7 0.0003	0	11 0.0004	0	123 0.0047	0	9 0.0003	0	126 0.0048	13 0.0005	0	3 0.0001	1 0	11 0.0004	0	0	0	6 0.0002	3 0.0001	25853 0.9818	0	3 0.0001	5 0.0002	0	1 0	0.9818
phacoemulsifier handpiece & Bonn forceps	13 0.0171	0	0	0	0	0	0	0	1 0.0013	1 0.0013	6 0.0079	29 0.0381	0	0	0	0	0	0	0	0	0	0	712 0.9344	0	0	0	0	0.9344
phacoemulsifier handpiece & micromanipulator	492 0.009	0	74 0.0014	5 0.0001	36 0.0007	0	15 0.0003	0	9 0.0002	8 0.0001	506 0.0092	291 0.0053	30 0.0005	0	0	53 0.001	0	0	0	0	0	1 0	0	53239 0.9717	28 0.0005	4 0.0001	0	0.9717
irrigation/aspiration handpiece & micromanipulator	189 0.0143	0	13 0.001	3 0.0002	61 0.0046	0	33 0.0025	0	9 0.0007	0	817 0.0619	8 0.0006	5 0.0004	1 0.0001	1 0.0001	30 0.0023	0	0	0	1 0.0001	2 0.0002	3 0.0002	0	174 0.0132	11846 0.8976	1 0.0001	0	0.8976
hydrodissection cannula & micromanipulator	7 0.0033	0	14 0.0066	0	0	0	0	0	1 0.0005	0	1 0.0005	0	0	0	0	8 0.0038	1 0.0005	0	0	0	0	0	0	3 0.0014	0	2097 0.9836	0	0.9836
Troutman forceps & suture needle	41 0.0321	0	1 0.0008	0	0	0	4 0.0031	0	1 0.0008	1 0.0008	18 0.0141	1 0.0008	1 0.0008	0	0	3 0.0023	16 0.0125	0	0	2 0.0016	0	2 0.0016	0	0	0	0	1186 0.9287	0.9287
Precision	0.9668	0.908	0.9487	0.9417	0.9461	0.9685	0.9785	0.9143	0.977	0.9657	0.9776	0.9488	0.9707	0.958	0.9221	0.9506	0.9479	0.981	1	0.9177	0.9629	0.9966	0.9596	0.9881	0.9798	0.9887	0.9549	

Table A.2: Confusion matrix rounded to 4 digits of the workflow augmented test data for the LSTM classifier.

Prediction Label	no tool in contact	biomarker	hydrodissection cannula	Rycroft cannula	viscoelastic cannula	cotton	capsulorhexis cystotome	Bonn forceps	capsulorhexis forceps	Troutman forceps	irrigation/aspiration handpiece	phacoemulsifier handpiece	implant injector	primary incision knife	secondary incision knife	micromanipulator	suture needle	Mendez ring	Mendez ring & biomarker	Bonn forceps & secondary incision knife	primary incision knife & Bonn forceps	capsulorhexis cystotome & Bonn forceps	phacoemulsifier handpiece & Bonn forceps	phacoemulsifier handpiece & micromanipulator	irrigation/aspiration handpiece & micromanipulator	hydrodissection cannula & micromanipulator	Troutman forceps & suture needle	Recall
no tool in contact	480560 0.9908	27 0.0001	291 0.0006	311 0.0006	659 0.0014	186 0.0004	446 0.0009	64 0.0001	63 0.0001	177 0.0004	956 0.002	119 0.0002	189 0.0004	72 0.0001	273 0.0006	424 0.0009	26 0.0001	77 0.0002	0	47 0.0001	9 0	0	0	43 0.0001	12 0	1 0	12 0	0.9908
biomarker	423 0.4942	427 0.4988	0	0	1 0.0012	1 0.0012	0	0	0	0	0	0	0	1 0.0012	2 0.0023	0	0	1 0.0012	0	0	0	0	0	0	0	0	0	0.4988
hydrodissection cannula	948 0.0478	1 0.0001	18305 0.9239	0	142 0.0072	0	241 0.0122	0	17 0.0009	1 0.0001	59 0.003	39 0.002	0	7 0.0004	3 0.0002	25 0.0013	0	0	0	0	0	2 0.0001	0	14 0.0007	0	9 0.0005	0	0.9239
Rycroft cannula	1606 0.2154	0	2 0.0003	5584 0.7488	37 0.005	3 0.0004	0	0	0	0	147 0.0197	0	9 0.0012	0	0	58 0.0078	0	0	0	0	0	0	0	1 0.0001	10 0.0013	0	0	0.7488
viscoelastic cannula	1700 0.0627	0	35 0.0013	15 0.0006	24950 0.92	0	176 0.0065	0	6 0.0002	1 0	95 0.0035	1 0	5 0.0002	11 0.0004	3 0.0001	112 0.0041	0	0	0	0	1 0	1 0	0	6 0.0002	3 0.0001	0	0	0.9200
cotton	3423 0.2417	0	0	0	0	10735 0.7581	0	0	0	0	0	0	1 0.0001	0	0	2 0.0001	0	0	0	0	0	0	0	0	0	0	0	0.7581
capsulorhexis cystotome	1298 0.027	0	103 0.0021	1 0	195 0.0041	1 0	46299 0.9637	0	28 0.0006	0	80 0.0017	2 0	1 0	5 0.0001	5 0.0001	11 0.0002	0	0	0	0	0	13 0.0003	0	1 0	0	1 0	0	0.9637
Bonn forceps	383 0.184	0	0	0	6 0.0029	0	0	1595 0.7665	0	0	0	0	0	0	1 0.0005	0	0	0	0	49 0.0235	42 0.0202	1 0.0005	4 0.0019	0	0	0	0	0.7665
capsulorhexis forceps	288 0.0217	0	15 0.0011	0	9 0.0007	0	114 0.0086	2 0.0002	12796 0.9659	1 0.0001	14 0.0011	2 0.0002	2 0.0002	3 0.0002	0	0	0	0	0	1 0.0001	0	0	0	1 0.0001	0	0	0	0.9659
Troutman forceps	744 0.0972	0	1 0.0001	4 0.0005	2 0.0003	0	0	0	1 0.0001	6822 0.8912	28 0.0037	2 0.0003	34 0.0044	0	0	14 0.0018	0	0	0	0	0	0	0	0	0	1 0.0001	2 0.0003	0.8912
irrigation/aspiration handpiece	3121 0.0184	0	46 0.0003	26 0.0002	162 0.001	0	91 0.0005	5 0	56 0.0003	11 0.0001	165342 0.9745	96 0.0006	82 0.0005	20 0.0001	3 0	330 0.0019	0	0	0	0	6 0	3 0	1 0	48 0.0003	215 0.0013	0	1 0	0.9745
irrigation/aspiration handpiece	1060 0.0453	0	296 0.0127	0	24 0.001	0	50 0.0021	3 0.0001	18 0.0008	4 0.0002	259 0.0111	21459 0.9174	28 0.0012	2 0.0001	9 0.0004	4 0.0002	0	0	0	4 0.0002	0	7 0.0003	28 0.0012	136 0.0058	0	1 0	0	0.9174
implant injector	679 0.0404	0	0	17 0.001	0	0	0	0	0	15 0.0009	193 0.0115	6 0.0004	15847 0.9435	1 0.0001	0	37 0.0022	0	0	0	0	1 0.0001	0	0	0	0	0	0	0.9435
primary incision knife	504 0.1125	0	11 0.0025	0	44 0.0098	0	25 0.0056	0	4 0.0009	0	12 0.0027	11 0.0025	7 0.0016	3738 0.8342	15 0.0033	0	0	0	0	8 0.0018	102 0.0228	0	0	0	0	0	0	0.8342
secondary incision knife	758 0.1702	0	3 0.0007	0	7 0.0016	0	6 0.0013	0	0	0	1 0.0002	9 0.002	0	10 0.0022	3615 0.8116	0	0	0	0	43 0.0097	0	2 0.0004	0	0	0	0	0	0.8116
micromanipulator	987 0.046	0	1 0	14 0.0007	106 0.0049	0	0	0	0	8 0.0004	324 0.0151	0	31 0.0014	0	0	19971 0.9304	0	0	0	0	0	0	0	9 0.0004	8 0.0004	6 0.0003	0	0.9304
suture needle	239 0.2163	0	0	1 0.0009	1 0.0009	0	0	0	0	0	16 0.0145	0	2 0.0018	1 0.0009	0	8 0.0072	802 0.7258	0	0	0	0	0	0	0	0	0	35 0.0317	0.7258
Mendez ring	291 0.0623	0	1 0.0002	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4378 0.9375	0	0	0	0	0	0	0	0	0	0.9375
Mendez ring & biomarker	1 0.0244	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40 0.9756	0	0	0	0	0	0	0	0	0.9756
Bonn forceps & secondary incision knife	244 0.0787	0	1 0.0003	0	7 0.0023	0	4 0.0013	55 0.0177	3 0.001	0	3 0.001	3 0.001	2 0.0006	13 0.0042	49 0.0158	0	0	0	0	2701 0.8707	9 0.0029	6 0.0019	2 0.0006	0	0	0	0	0.8707
primary incision knife & Bonn forceps	196 0.0345	0	4 0.0007	0	17 0.003	3 0.0005	7 0.0012	109 0.0192	3 0.0005	1 0.0002	4 0.0007	2 0.0004	1 0.0002	126 0.0222	2 0.0004	0	0	0	0	13 0.0023	5186 0.9121	6 0.0011	5 0.0009	1 0.0002	0	0	0	0.9121
capsulorhexis cystotome & Bonn forceps	73 0.0028	0	12 0.0005	0	12 0.0005	0	216 0.0082	0	3 0.0001	0	23 0.0009	2 0.0001	0	0	0	1 0	0	1 0	0	5 0.0002	0	25975 0.9864	5 0.0002	3 0.0001	2 0.0001	0	0	0.986
phacoemulsifier handpiece & Bonn forceps	7 0.0092	0	0	0	0	3 0.0039	0	2 0.0026	1 0.0013	2 0.0026	6 0.0079	42 0.0551	0	0	0	0	0	0	0	1 0.0013	2 0.0026	0	696 0.9134	0	0	0	0	0.9134
phacoemulsifier handpiece & micromanipulator	795 0.0145	0	248 0.0045	0	178 0.0032	0	144 0.0026	0	9 0.0002	6 0.0001	221 0.004	436 0.008	32 0.0006	2 0	2 0	87 0.0016	0	0	0	3 0.0001	1 0	1 0	0	52553 0.9592	69 0.0013	4 0.0001	0	0.9592
irrigation/aspiration handpiece & micromanipulator	136 0.0103	0	0	5 0.0004	18 0.0014	0	1 0.0001	0	1 0.0001	4 0.0003	962 0.0729	1 0.0001	0	0	0	66 0.005	0	0	0	0	0	0	0	31 0.0023	11970 0.907	2 0.0002	0	0.9070
hydrodissection cannula & micromanipulator	9 0.0042	0	8 0.0038	0	5 0.0023	0	0	0	0	0	1 0.0005	0	0	0	0	9 0.0042	0	0	0	0	0	0	0	0	1 0.0005	2099 0.9845	0	0.9845
Troutman forceps & suture needle	76 0.0595	0	0	0	3 0.0023	0	0	0	0	1 0.0008	20 0.0157	0	4 0.0031	0	0	0	13 0.0102	0	0	0	1 0.0008	0	0	0	1 0.0008	0	1158 0.9068	0.9068
Precision	0.9601	0.9385	0.9444	0.9341	0.9385	0.982	0.9682	0.8692	0.9836	0.9671	0.9797	0.9652	0.9736	0.9317	0.9078	0.9439	0.9536	0.9823	1	0.9395	0.9675	0.9984	0.9393	0.9944	0.9739	0.9882	0.9586	

Table A.3: Confusion matrix rounded to 4 digits of the real surgical videos for the workflow augmented trained CNN classifier.

Prediction Label	no tool in contact	biomarker	hydrodissection cannula	Rycroft cannula	viscoelastic cannula	cotton	capsulorhexis cystotome	Bonn forceps	capsulorhexis forceps	Troutman forceps	irrigation/aspiration handpiece	phacoemulsifier handpiece	implant injector	primary incision knife	secondary incision knife	micromanipulator	suture needle	Mendez ring	Mendez ring & biomarker	Bonn forceps & secondary incision knife	primary incision knife & Bonn forceps	capsulorhexis cystotome & Bonn forceps	phacoemulsifier handpiece & Bonn forceps	phacoemulsifier handpiece & micromanipulator	irrigation/aspiration handpiece & micromanipulator	hydrodissection cannula & micromanipulator	Troutman forceps & suture needle	Recall
no tool in contact	11284 0.9854	0	3 0.0003	17 0.0015	30 0.0026	0	5 0.00044	0	3 0.0003	0	55 0.0048	13 0.0011	8 0.0007	3 0.0007	7 0.0006	20 0.0017	0	0	0	2 0.0002	0	0	0	0	1 0.0001	0	0	0.9854
biomarker	11 0.8462	0	0	0	0	0	0	0	0	0	0	0	0	0	2 0.1538	0	0	0	0	0	0	0	0	0	0	0	0	0
hydrodissection cannula	49 0.0928	0	422 0.7992	0	9 0.017	0	1 0.0019	0	0	0	19 0.036	7 0.0133	0	0	0	19 0.036	0	0	0	0	0	0	0	2 0.0038	0	0	0	0.7992
Rycroft cannula	456 0.4706	0	8 0.0083	375 0.387	64 0.066	0	6 0.0062	0	0	1 0.001	21 0.0217	1 0.001	10 0.0103	0	1 0.001	22 0.0227	0	0	0	2 0.0021	1 0.001	0	0	1 0.001	0	0	0	0.387
viscoelastic cannula	62 0.1396	0	1 0.0023	1 0.0023	369 0.8311	0	1 0.0023	0	1 0.0023	0	3 0.0068	0	0	0	0	6 0.0135	0	0	0	0	0	0	0	0	0	0	0	0.8311
cotton	3 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
capsulorhexis cystotome	194 0.1152	0	19 0.0113	2 0.0012	92 0.0546	0	1242 0.7375	0	11 0.0065	0	66 0.0392	1 0.0006	9 0.0053	0	0	46 0.027	0	0	0	0	0	2 0.0012	0	0	0	0	0	0.7375
Bonn forceps	5 0.625	0	0	0	0	0	0	0	0	0	0	0	0	1 0.125	0	0	0	0	0	2 0.25	0	0	0	0	0	0	0	0
capsulorhexis forceps	24 0.1558	0	1 0.0065	0	1 0.0065	0	2 0.013	0	73 0.474	0	42 0.2727	3 0.0195	4 0.026	0	0	4 0.026	0	0	0	0	0	0	0	0	0	0	0	0.474
Troutman forceps	2 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
irrigation/aspiration handpiece	220 0.0466	0	13 0.0028	1 0.0002	4 0.0008	0	5 0.0011	0	5 0.0011	0	4428 0.9377	14 0.003	8 0.0017	0	0	8 0.00169	0	0	0	0	0	0	0	5 0.0011	11 0.0023	0	0	0.9377
phacoemulsifier handpiece	248 0.2165	0	13 0.0113	1 0.0009	0	0	4 0.0035	0	0	2 0.0017	113 0.0986	694 0.6056	28 0.0244	1 0.0009	0	2 0.0017	0	0	0	0	0	0	0	40 0.0349	0	0	0	0.6056
implant injector	46 0.1055	0	0	0	0	0	3 0.0069	0	0	1 0.0023	13 0.0298	0	370 0.8486	0	0	3 0.0069	0	0	0	0	0	0	0	0	0	0	0	0.8486
primary incision knife	34 0.2061	0	0	0	1 0.0061	0	1 0.0061	0	0	0	4 0.0242	1 0.0061	1 0.0061	94 0.567	0	1 0.0061	0	0	0	2 0.0121	26 0.1576	0	0	0	0	0	0	0.5697
secondary incision knife	56 0.5234	0	0	0	0	0	0	0	0	0	0	0	0	0	39 0.3645	0	0	0	0	12 0.1121	0	0	0	0	0	0	0	0.3645
micromanipulator	69 0.1624	0	5 0.0118	0	23 0.0541	0	15 0.0353	0	1 0.0024	0	4 0.0094	3 0.0071	3 0.0071	0	0	299 0.7035	0	0	0	0	0	0	0	3 0.0071	0	0	0	0.7035
suture needle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mendez ring	35 0.9211	0	0	0	0	0	0	1 0.0263	0	0	0	0	0	0	0	0	0	0	0	0	2 0.0526	0	0	0	0	0	0	0
Mendez ring & biomaker	2 0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2 0.5	0	0	0	0	0	0	0
Bonn forceps & secondary incision knife	9 0.1233	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64 0.8767	0	0	0	0	0	0	0	0.8767
primary incision knife & Bonn forceps	8 0.16	0	0	0	0	0	0	0	0	0	0	0	0	9 0.18	3 0.06	0	0	0	0	5 0.1	25 0.5	0	0	0	0	0	0	0.5
capsulorhexis cystotome & Bonn forceps	2 0.25	0	0	0	4 0.5	0	2 0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
phacoemulsifier handpiece & Bonn forceps	0	0	0	0	0	0	0	0	2 0.0541	1 0.027	1 0.027	20 0.5405	4 0.1081	0	0	3 0.081	0	0	0	0	0	0	6 0.1622	0	0	0	0	0.1622
phacoemulsifier handpiece & micromanipulator	966 0.2334	0	28 0.0068	3 0.0007	43 0.0104	0	19 0.0046	0	0	0	246 0.0594	30 0.0072	7 0.0017	0	0	52 0.0126	0	0	0	0	0	2 0.0005	0	2666 0.6443	75 0.0181	1 0.0002	0	0.6443
irrigation/aspiration handpiece & micromanipulator	56 0.0441	0	1 0.0008	0	14 0.011	0	0	0	1 0.0008	0	642 0.5059	0	0	0	0	4 0.0032	0	0	0	0	0	0	0	110 0.0867	441 0.3475	0	0	0.3475
hydrodissection cannula & micromanipulator	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Troutman forceps & suture needle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Precision	0.8153	n/a	0.8210	0.9375	0.5642	n/a	0.951	0	0.7526	0	0.7827	0.8818	0.8186	0.8704	0.75	0.6115	n/a	n/a	n/a	0.7191	0.4464	0	1	0.943	0.8352	0	n/a	

Table A.4: Confusion matrix rounded to 4 digits of the real surgical videos for the workflow augmented trained LSTM classifier.

Prediction Label	no tool in contact	biomarker	hydrodissection cannula	Rycroft cannula	viscoelastic cannula	cotton	capsulorhexis cystotome	Bonn forceps	capsulorhexis forceps	Troutman forceps	irrigation/aspiration handpiece	phacoemulsifier handpiece	implant injector	primary incision knife	secondary incision knife	micromanipulator	suture needle	Mendez ring	Mendez ring & biomarker	Bonn forceps & secondary incision knife	primary incision knife & Bonn forceps	capsulorhexis cystotome & Bonn forceps	phacoemulsifier handpiece & Bonn forceps	phacoemulsifier handpiece & micromanipulator	irrigation/aspiration handpiece & micromanipulator	hydrodissection cannula & micromanipulator	Troutman forceps & suture needle	Recall
no tool in contact	11223 0.9801	0	11 0.0009	69 0.006	17 0.0015	0	20 0.0017	0	1 0.0001	0	60 0.0052	23 0.002	4 0.0003	0	6 0.0005	14 0.0012	0	0	0	1 0.0001	0	0	0	2 0.0002	0	0	0	0.9801
biomarker	12 0.9231	0	0	0	0	0	0	0	0	0	0	0	0	0	1 0.0769	0	0	0	0	0	0	0	0	0	0	0	0	0
hydrodissection cannula	31 0.0587	0	488 0.9242	0	0	0	8 0.0152	0	0	0	0	1 0.0019	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9242
Rycroft cannula	217 0.2239	0	0	705 0.7276	9 0.0093	0	0	0	0	0	25 0.0258	0	0	0	0	13 0.0134	0	0	0	0	0	0	0	0	0	0	0	0.7276
viscoelastic cannula	46 0.1036	0	1 0.0023	0	386 0.8694	0	11 0.0248	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8694
cotton	3 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
capsulorhexis cystotome	39 0.0232	0	5 0.003	0	14 0.0083	0	1615 0.959	0	0	0	0	0	0	1 0.0006	0	0	0	0	0	0	0	0	0	10 0.0059	0	0	0	0.959
Bonn forceps	8 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
capsulorhexis forceps	19 0.1234	0	8 0.0519	0	0	0	26 0.1688	0	99 0.6429	0	1 0.0065	0	0	0	0	0	0	0	0	0	0	1 0.0065	0	0	0	0	0	0.6429
Troutman forceps	1 0.5	0	0	0	0	0	1 0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
irrigation/aspiration handpiece	51 0.0108	0	0	0	0	0	0	0	0	0	4645 0.9837	2 0.0004	0	0	0	5 0.0011	0	0	0	0	0	0	0	6 0.0013	13 0.0028	0	0	0.9837
phacoemulsifier handpiece	33 0.0288	0	5 0.0044	0	0	0	0	0	0	0	12 0.0105	1068 0.9319	0	0	0	0	0	0	0	0	0	0	0	28 0.0244	0	0	0	0.9319
implant injector	31 0.0711	0	0	0	0	0	0	0	0	0	16 0.0367	0	382 0.8761	0	0	7 0.0161	0	0	0	0	0	0	0	0	0	0	0	0.8761
primary incision knife	36 0.2182	0	5 0.0303	0	1 0.0061	0	0	0	0	0	0	1 0.0061	0	105 0.6364	1 0.0061	0	0	0	0	0	16 0.097	0	0	0	0	0	0	0.6364
secondary incision knife	47 0.4393	0	0	0	0	0	0	0	0	0	0	0	0	0	48 0.4486	0	0	0	0	12 0.1121	0	0	0	0	0	0	0	0.4486
micromanipulator	40 0.0941	0	0	0	10 0.0235	0	0	0	0	0	6 0.0141	0	2 0.0047	0	0	359 0.8447	0	0	0	0	0	0	0	8 0.0188	0	0	0	0.8447
suture needle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mendez ring	36 0.9474	0	0	0	0	0	0	0	0	0	1 0.0263	0	0	0	0	0	0	1 0.0263	0	0	0	0	0	0	0	0	0	0.0263
Mendez ring & biomarker	3 0.75	0	0	0	0	0	0	0	0	0	0	0	1 0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bonn forceps & secondary incision knife	16 0.2192	0	0	0	0	0	0	1 0.0137	0	0	0	0	0	1 0.0137	1 0.0137	0	0	0	0	54 0.7397	0	0	0	0	0	0	0	0.7397
primary incision knife & Bonn forceps	5 0.1	0	0	0	0	0	0	0	0	0	0	0	0	20 0.4	0	0	0	0	0	1 0.02	24 0.48	0	0	0	0	0	0	0.48
capsulorhexis cystotome & Bonn forceps	0	0	0	0	0	0	8 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
phacoemulsifier handpiece & Bonn forceps	0	0	0	0	0	0	0	0	0	0	0	36 0.973	0	0	0	0	0	0	0	0	0	0	1 0.027	0	0	0	0	0.027
phacoemulsifier handpiece & micromanipulator	3 0.0007	0	1 0.0002	0	0	0	0	0	0	0	35 0.0085	18 0.0043	0	0	0	0	0	0	0	0	0	0	0	4034 0.9749	47 0.0114	0	0	0.9749
irrigation/aspiration handpiece & micromanipulator	7 0.0055	0	0	0	0	0	0	0	0	0	286 0.2254	0	0	0	0	1 0.0008	0	0	0	0	0	0	0	153 0.1206	822 0.6478	0	0	0.6478
hydrodissection cannula & micromanipulator	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Troutman forceps & suture needle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Precision	0.9426	n/a	0.9313	0.91085	0.8833	n/a	0.9562	0	0.99	n/a	0.9131	0.9295	0.9820	0.8268	0.8421	0.8997	n/a	1	n/a	0.7941	0.6	0	1	0.9512	0.932	n/a	n/a	

Table A.5: Confusion matrix rounded to 4 digits of the real surgical videos for the split trained CNN Classifier.

Prediction Label	no tool in contact	biomarker	hydrodissection cannula	Rycroft cannula	viscoelastic cannula	cotton	capsulorhexis cystotome	Bonn forceps	capsulorhexis forceps	Troutman forceps	irrigation/aspiration handpiece	phacoemulsifier handpiece	implant injector	primary incision knife	secondary incision knife	micromanipulator	suture needle	Mendez ring	Mendez ring & biomarker	Bonn forceps & secondary incision knife	primary incision knife & Bonn forceps	capsulorhexis cystotome & Bonn forceps	phacoemulsifier handpiece & Bonn forceps	phacoemulsifier handpiece & micromanipulator	irrigation/aspiration handpiece & micromanipulator	hydrodissection cannula & micromanipulator	Troutman forceps & suture needle	Recall
no tool in contact	11001 0.9607	0	10 0.0009	78 0.0068	21 0.0018	0	4 0.0003	0	4 0.0003	4 0.0003	115 0.01	90 0.0079	10 0.0009	6 0.0005	4 0.0003	31 0.0027	0	1 0.0001	0	4 0.0003	0	0	0	62 0.0054	6 0.0005	0	0	0.9607
biomarker	13 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hydrodissection cannula	61 0.1155	0	260 0.4924	10 0.0189	6 0.0114	0	2 0.0038	0	1 0.0019	0	85 0.161	29 0.0549	0	0	0	32 0.0606	0	0	0	0	0	0	0	42 0.0795	0	0	0	0.4924
Rycroft cannula	326 0.3364	0	2 0.0021	511 0.5273	53 0.0547	0	10 0.0103	0	0	0	6 0.0062	1 0.001	3 0.0031	0	1 0.001	6 0.0062	0	0	0	0	0	0	0	42 0.0433	8 0.0083	0	0	0.5273
viscoelastic cannula	53 0.1194	0	5 0.0113	16 0.036	328 0.7387	0	15 0.0338	0	0	0	13 0.0293	0	0	2 0.0045	0	10 0.0225	0	0	0	0	0	0	0	1 0.0023	0	1 0.0023	0	0.7387
cotton	3 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
capsulorhexis cystotome	241 0.1431	0	66 0.0392	13 0.0077	209 0.1241	0	879 0.522	0	10 0.0059	0	111 0.0659	3 0.0018	14 0.0083	1 0.0006	0	104 0.0618	0	0	0	0	0	1 0.0006	0	12 0.0071	20 0.0119	0	0	0.522
Bonn forceps	4 0.5	0	0	1 0.125	0	0	0	0	0	0	0	0	0	0	1 0.125	0	0	0	0	1 0.125	1 0.125	0	0	0	0	0	0	0
capsulorhexis forceps	35 0.2273	0	3 0.0195	0	6 0.039	0	2 0.013	0	18 0.1169	0	83 0.539	1 0.0065	1 0.0065	0	0	4 0.026	0	0	0	1 0.0065	0	0	0	0	0	0	0	0.1169
Troutman forceps	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2 1	0	0	0	0	0	0	0	0	0	0	0	0
irrigation/aspiration handpiece	307 0.065	0	71 0.015	17 0.0036	66 0.014	0	25 0.0053	0	6 0.0013	0	3767 0.7978	180 0.0381	24 0.0051	0	0	38 0.008	0	0	0	0	0	0	7 0.0015	189 0.04	24 0.0051	1 0.0002	0	0.7978
phacoemulsifier handpiece	196 0.171	0	38 0.0332	3 0.0026	1 0.0009	0	5 0.0044	0	2 0.0017	0	228 0.199	536 0.4677	31 0.0271	2 0.0017	0	0	0	0	0	0	0	0	6 0.0052	98 0.0855	0	0	0	0.4677
implant injector	37 0.0849	0	0	5 0.0115	0	0	12 0.0275	0	0	2 0.0046	8 0.0183	20 0.0459	341 0.7821	4 0.0092	0	4 0.0092	0	0	0	0	1 0.0023	0	0	2 0.0046	0	0	0	0.7821
primary incision knife	27 0.1636	0	1 0.0061	7 0.0424	3 0.0182	0	0	0	0	0	9 0.0545	6 0.0364	11 0.0667	63 0.3818	2 0.0121	0	0	0	0	2 0.0121	33 0.2	1 0.0061	0	0	0	0	0	0.3818
secondary incision knife	46 0.4299	0	0	16 0.1495	0	0	0	0	0	0	3 0.028	0	2 0.0187	0	32 0.2991	0	0	0	0	7 0.0654	0	0	0	1 0.0093	0	0	0	0.2991
micromanipulator	52 0.1224	0	37 0.0871	14 0.0329	26 0.0612	0	44 0.1035	0	1 0.0024	1 0.0024	2 0.0047	8 0.0188	7 0.0165	0	1 0.0024	217 0.5106	0	0	0	0	0	1 0.0024	0	10 0.0235	1 0.0024	3 0.0071	0	0.5106
suture needle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mendez ring	38 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mendez ring & biomaker	4 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bonn forceps & secondary incision knife	29 0.3973	0	0	2 0.0274	0	0	0	0	0	0	3 0.0411	0	0	0	2 0.0274	0	0	0	0	36 0.4932	0	0	0	0	1 0.0137	0	0	0.4932
primary incision knife & Bonn forceps	9 0.18	0	0	3 0.06	1 0.02	0	0	0	0	0	0	0	0	4 0.08	0	0	0	0	0	4 0.08	21 0.42	0	6 0.12	2 0.04	0	0	0	0.42
capsulorhexis cystotome & Bonn forceps	0	0	0	0	6 0.75	0	2 0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
phacoemulsifier handpiece & Bonn forceps	1 0.027	0	0	0	0	0	0	0	2 0.0541	3 0.0811	8 0.2162	4 0.1081	2 0.0541	0	0	0	0	0	0	0	0	0	4 0.1081	12 0.3243	1 0.027	0	0	0.1081
phacoemulsifier handpiece & micromanipulator	644 0.1556	0	74 0.0179	73 0.0176	125 0.0302	0	24 0.0058	0	1 0.0002	2 0.0005	566 0.1368	87 0.021	26 0.0063	0	0	67 0.0162	0	0	0	0	1 0.0002	1 0.0002	2 0.0005	2248 0.5433	196 0.0474	1 0.0002	0	0.5433
irrigation/aspiration handpiece & micromanipulator	71 0.0559	0	8 0.0063	4 0.0032	25 0.0197	0	5 0.0039	0	2 0.0016	2 0.0016	209 0.1647	5 0.0039	4 0.0032	1 0.0008	0	20 0.0158	0	0	0	0	0	0	1 0.0008	400 0.3152	510 0.4019	2 0.0016	0	0.4019
hydrodissection cannula & micromanipulator	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Troutman forceps & suture needle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Precision	0.8335	n/a	0.4522	0.6611	0.3744	n/a	0.8542	n/a	0.383	0	0.7222	0.5526	0.7164	0.7590	0.7442	0.4056	n/a	0	n/a	0.6545	0.3684	0	0.1538	0.7203	0.6649	0	n/a	

Table A.6: Confusion matrix rounded to 4 digits of the real surgical videos for the split trained LSTM classifier.

Prediction Label	no tool in contact	biomarker	hydrodissection cannula	Rycroft cannula	viscoelastic cannula	cotton	capsulorhexis cystotome	Bonn forceps	capsulorhexis forceps	Troutman forceps	irrigation/aspiration handpiece	phacoemulsifier handpiece	implant injector	primary incision knife	secondary incision knife	micromanipulator	suture needle	Mendez ring	Mendez ring & biomarker	Bonn forceps & secondary incision knife	primary incision knife & Bonn forceps	capsulorhexis cystotome & Bonn forceps	phacoemulsifier handpiece & Bonn forceps	phacoemulsifier handpiece & micromanipulator	irrigation/aspiration handpiece & micromanipulator	hydrodissection cannula & micromanipulator	Troutman forceps & suture needle	Recall
no tool in contact	11108 0.97	0	14 0.0012	130 0.0114	12 0.001	0	17 0.0015	0	14 0.0012	0	40 0.0035	9 0.0008	13 0.0011	12 0.001	6 0.0005	67 0.0059	0	0	0	0	1 0.0001	0	0	8 0.0007	0	0	0	0.97
biomarker	11 0.8462	0	0	0	0	0	0	0	0	0	0	0	0	1 0.0769	1 0.0769	0	0	0	0	0	0	0	0	0	0	0	0	0
hydrodissection cannula	39 0.0739	0	435 0.8239	4 0.0076	6 0.0114	0	28 0.053	0	0	0	0	4 0.0076	3 0.0057	2 0.0038	0	6 0.0114	0	0	0	0	0	0	0	1 0.0019	0	0	0	0.8239
Rycroft cannula	155 0.16	0	0	784 0.8091	5 0.0052	0	0	0	0	0	2 0.0021	0	0	0	0	23 0.0237	0	0	0	0	0	0	0	0	0	0	0	0.8091
viscoelastic cannula	46 0.1036	0	0	6 0.0135	325 0.732	0	30 0.0676	0	0	0	1 0.0023	0	0	0	0	34 0.0766	0	0	0	0	0	0	0	2 0.0045	0	0	0	0.732
cotton	3 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
capsulorhexis cystotome	86 0.0511	0	39 0.0232	2 0.0012	39 0.0232	0	1508 0.8955	0	4 0.0024	0	0	0	0	1 0.0006	0	5 0.003	0	0	0	0	0	0	0	0	0	0	0	0.8955
Bonn forceps	5 0.625	0	0	0	0	0	0	0	0	0	0	0	0	2 0.25	1 0.125	0	0	0	0	0	0	0	0	0	0	0	0	0
capsulorhexis forceps	30 0.1948	0	9 0.0584	0	1 0.0065	0	54 0.3506	0	52 0.3377	0	1 0.0065	2 0.013	0	0	0	3 0.0195	0	0	0	0	0	0	0	0	2 0.013	0	0	0.3377
Troutman forceps	0	0	1 0.5	1 0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
irrigation/aspiration handpiece	74 0.0157	0	1 0.0002	50 0.0106	13 0.0028	0	1 0.0002	0	1 0.0002	0	4550 0.9636	4 0.0008	1 0.0002	0	0	7 0.0015	0	0	0	0	0	0	0	6 0.0013	14 0.003	0	0	0.9636
phacoemulsifier handpiece	56 0.0489	0	10 0.0087	2 0.0017	1 0.0009	0	0	0	1 0.0009	0	68 0.0593	945 0.8246	21 0.0183	2 0.0017	0	0	0	0	0	0	0	0	0	39 0.034	1 0.0009	0	0	0.8246
implant injector	34 0.078	0	1 0.0023	22 0.0505	0	0	0	0	1 0.0023	0	5 0.0115	0	371 0.8509	0	0	2 0.0046	0	0	0	0	0	0	0	0	0	0	0	0.8509
primary incision knife	33 0.2	0	3 0.0182	1 0.0061	4 0.0242	0	1 0.0061	0	1 0.0061	0	0	5 0.0303	6 0.0364	111 0.6727	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6727
secondary incision knife	36 0.3364	0	0	1 0.0093	0	0	0	0	0	0	0	0	0	10 0.0935	60 0.5607	0	0	0	0	0	0	0	0	0	0	0	0	0.5607
micromanipulator	32 0.0753	0	2 0.0047	104 0.2447	28 0.0659	0	2 0.0047	0	0	0	3 0.0071	0	4 0.0094	0	0	240 0.5647	0	0	0	0	0	0	0	10 0.0235	0	0	0	0.5647
suture needle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mendez ring	35 0.9211	0	0	0	0	0	0	0	0	0	0	0	0	3 0.0789	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mendez ring & biomaker	3 0.75	0	0	0	0	0	0	0	1 0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bonn forceps & secondary incision knife	12 0.1644	0	0	0	0	0	0	0	0	0	0	0	0	11 0.1507	34 0.4658	0	0	0	0	16 0.2192	0	0	0	0	0	0	0	0.2192
primary incision knife & Bonn forceps	10 0.2	0	0	0	0	0	0	0	0	0	0	0	0	36 0.72	4 0.08	0	0	0	0	0	0	0	0	0	0	0	0	0
capsulorhexis cystotome & Bonn forceps	0	0	0	0	0	0	8 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
phacoemulsifier handpiece & Bonn forceps	1 0.027	0	0	0	0	0	0	0	3 0.0811	0	0	28 0.7568	2 0.0541	0	0	0	0	0	0	0	0	0	0	3 0.0811	0	0	0	0
phacoemulsifier handpiece & micromanipulator	16 0.0039	0	0	2 0.0005	0	0	0	0	0	0	30 0.0072	22 0.0053	0	0	0	0	0	0	0	0	0	0	0	3906 0.9439	162 0.0391	0	0	0.9439
irrigation/aspiration handpiece & micromanipulator	3 0.0024	0	0	26 0.0205	8 0.0063	0	0	0	1 0.0008	0	183 0.1442	0	0	0	0	1 0.0008	0	0	0	0	0	0	0	130 0.1024	917 0.7226	0	0	0.7226
hydrodissection cannula & micromanipulator	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Troutman forceps & suture needle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Precision	0.9391	n/a	0.8447	0.6907	0.7353	n/a	0.9145	n/a	0.6582	n/a	0.9318	0.9274	0.8812	0.5812	0.566	0.6186	n/a	n/a	n/a	1	0	n/a	n/a	0.9515	0.8367	n/a	n/a	

Appendix **B**

Project: Post-operative Image Prediction

B.1 Implementation Details to Calculate S_{mod}

In this section, we describe the optimization algorithm to automatically find the shape modifications S_{mod} in more detail. Based on the statistical point distribution model (BFM2009) by [140], the coordinates x_i (x,y, and z) of each vertex i of each 3D face S^n can be described by

$$x = V\alpha + \overline{x} \tag{B.1}$$

with V being a matrix of 199 eigenvectors to maximally describe the variance of faces, α being a parameter vector with 199 elements, and $\overline{x}$ being the mean face. To define the locally modified region, we manually selected a facial region, e.g., the nose or the chin and labeled all vertices x_i within the selected region to belong to $Mask$. Consequently, all other vertices were labeled to belong to $\overline{Mask}$. We aimed to find such an $\hat{\alpha}$ that maximally deflects all vertices within $Mask$ while minimally deflecting all other vertices within $\overline{Mask}$. To achieve this, we optimized the following objective:

$$\min_{\alpha} \sum_{x_i \in \overline{Mask}} \|x_i\|_F - \lambda_1 \sum_{x_i \in Mask} \|x_i\|_F + \lambda_2 \left(\|\alpha\|_F - 1\right)^2 \tag{B.2}$$

with $\|.\|_F$ being the Frobenius norm, $\lambda_1 = 4$ to control the deflection, and $\lambda_2 = 1000$ to regularize the solution $\hat{\alpha}$ to a constant length. Having such an optimized $\hat{\alpha}$, we were able to create a linearly scalable local modification S_{mod} using a scalar λ:

$$S_{mod} = \lambda\hat{\alpha} \tag{B.3}$$

As an example, we generated an enlarged nose modification by choosing $\lambda > 0$ and a shrunken nose modification by choosing $\lambda < 0$. On one hand, this approach to derive S_{mod} can generate local deflections on any region of the face as long as these deflections can be represented by the point distribution model. On the other hand, this approach has the disadvantage that it cannot generate specific local modifications since the optimization algorithm only focuses on a maximal deflection of the selected vertices.

B.2 Network Architectures

In the Tables B.1 to B.4, the detailed architectures are given for all neural networks of this study using the following abbreviations: CONV=2D convolutional layer, DECONV=2D transposed convolutional layer, BN=batch normalization, N=number of output channels, K=kernel size, S=stride size, P=padding size. The width and height are set to the image resolution, i.e., $h = 128$, $w = 128$ except for the local discriminator D_{Roi} where we set the width and height between 16 and 48. All leaky rectifying linear units (LeakyReLU) were implemented using a negative slope of 0.01.

Table B.1: Architecture of G

description	input shape →output shape	layer details
input layer	$(h, w, 6) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(N64, K4×4, S2, P1), LeakyReLU
down-sampling	$(\frac{h}{2}, \frac{w}{2}, 64) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(N128, K4×4, S2, P1), BN, LeakyReLU
down-sampling	$(\frac{h}{4}, \frac{w}{4}, 128) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(N256, K4×4, S2, P1), BN, LeakyReLU
down-sampling	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	CONV-(N512, K4×4, S2, P1), BN, LeakyReLU
down-sampling	$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	CONV-(N512, K4×4, S2, P1), BN, LeakyReLU
up-sampling	$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	DECONV-(N512, K4×4, S2, P1), BN, LeakyReLU
up-sampling	$(\frac{h}{16}, \frac{w}{16}, 1024) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	DECONV-(N256, K4×4, S2, P1), BN, LeakyReLU
up-sampling	$(\frac{h}{8}, \frac{w}{8}, 512) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	DECONV-(N64, K4×4, S2, P1), BN, LeakyReLU
up-sampling	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	DECONV-(N64, K4×4, S2, P1), BN, LeakyReLU
output layer	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (h, w, 3)$	DECONV-(N3, K4×4, S2, P1), Tanh

Table B.2: Architecture of D

description	input shape →output shape	layer details
input layer	$(h, w, 3) \rightarrow (\frac{h}{2}, \frac{w}{2}, 48)$	CONV-(N48, K4×4, S2, P1), LeakyReLU
hidden layer	$(\frac{h}{2}, \frac{w}{2}, 48) \rightarrow (\frac{h}{4}, \frac{w}{4}, 96)$	CONV-(N96, K4×4, S2, P1), LeakyReLU
hidden layer	$(\frac{h}{4}, \frac{w}{4}, 96) \rightarrow (\frac{h}{8}, \frac{w}{8}, 192)$	CONV-(N192, K4×4, S2, P1), LeakyReLU
hidden layer	$(\frac{h}{8}, \frac{w}{8}, 192) \rightarrow (\frac{h}{16}, \frac{w}{16}, 384)$	CONV-(N384, K4×4, S2, P1), LeakyReLU
hidden layer	$(\frac{h}{16}, \frac{w}{16}, 384) \rightarrow (\frac{h}{32}, \frac{w}{32}, 768)$	CONV-(N768, K4×4, S2, P1), LeakyReLU
hidden layer	$(\frac{h}{32}, \frac{w}{32}, 768) \rightarrow (\frac{h}{64}, \frac{w}{64}, 1536)$	CONV-(N1536, K4×4, S2, P1), LeakyReLU
hidden layer	$(\frac{h}{64}, \frac{w}{64}, 1536) \rightarrow (\frac{h}{64}, \frac{w}{64}, 1)$	CONV-(N1, K3×3, S2, P1), LeakyReLU
output layer	$(\frac{h}{64}, \frac{w}{64}, 1) \rightarrow (1)$	mean

Table B.3: Architecture of D_{Roi}

description	input shape →output shape	layer details
input layer	$(h, w, 3) \rightarrow (h, w, 48)$	CONV-(N48, K3×3, S1, P1), LeakyReLU
hidden layer	$(h, w, 48) \rightarrow (\frac{h}{2}, \frac{w}{2}, 96)$	CONV-(N96, K4×4, S2, P1), LeakyReLU
hidden layer	$(\frac{h}{2}, \frac{w}{2}, 96) \rightarrow (\frac{h}{2}, \frac{w}{2}, 192)$	CONV-(N192, K3×3, S1, P1), LeakyReLU
hidden layer	$(\frac{h}{2}, \frac{w}{2}, 192) \rightarrow (\frac{h}{4}, \frac{w}{4}, 384)$	CONV-(N384, K4×4, S2, P1), LeakyReLU
hidden layer	$(\frac{h}{4}, \frac{w}{4}, 384) \rightarrow (\frac{h}{4}, \frac{w}{4}, 768)$	CONV-(N768, K3×3, S1, P1), LeakyReLU
hidden layer	$(\frac{h}{4}, \frac{w}{4}, 768) \rightarrow (\frac{h}{8}, \frac{w}{8}, 1536)$	CONV-(N1536, K4×4, S2, P1), LeakyReLU
hidden layer	$(\frac{h}{8}, \frac{w}{8}, 1536) \rightarrow (\frac{h}{8}, \frac{w}{8}, 1)$	CONV-(N1, K3×3, S1, P1), LeakyReLU
output layer	$(\frac{h}{8}, \frac{w}{8}, 1) \rightarrow (1)$	mean

Table B.4: Architecture of G_S

description	input shape →output shape	layer details
input layer	$(h, w, 3) \rightarrow (\frac{h}{2}, \frac{w}{2}, 48)$	CONV-(N64, K4×4, S2, P1), LeakyReLU
down-sampling	$(\frac{h}{2}, \frac{w}{2}, 48) \rightarrow (\frac{h}{4}, \frac{w}{4}, 96)$	CONV-(N128, K4×4, S2, P1), BN, LeakyReLU
down-sampling	$(\frac{h}{4}, \frac{w}{4}, 96) \rightarrow (\frac{h}{8}, \frac{w}{8}, 192)$	CONV-(N256, K4×4, S2, P1), BN, LeakyReLU
down-sampling	$(\frac{h}{8}, \frac{w}{8}, 192) \rightarrow (\frac{h}{16}, \frac{w}{16}, 384)$	CONV-(N512, K4×4, S2, P1), BN, LeakyReLU
down-sampling	$(\frac{h}{16}, \frac{w}{16}, 384) \rightarrow (\frac{h}{32}, \frac{w}{32}, 384)$	CONV-(N512, K4×4, S2, P1), BN, LeakyReLU
up-sampling	$(\frac{h}{32}, \frac{w}{32}, 384) \rightarrow (\frac{h}{16}, \frac{w}{16}, 384)$	DECONV-(N512, K4×4, S2, P1), BN, LeakyReLU
up-sampling	$(\frac{h}{16}, \frac{w}{16}, 768) \rightarrow (\frac{h}{8}, \frac{w}{8}, 192)$	DECONV-(N256, K4×4, S2, P1), BN, LeakyReLU
up-sampling	$(\frac{h}{8}, \frac{w}{8}, 384) \rightarrow (\frac{h}{4}, \frac{w}{4}, 96)$	DECONV-(N64, K4×4, S2, P1), BN, LeakyReLU
up-sampling	$(\frac{h}{4}, \frac{w}{4}, 192) \rightarrow (\frac{h}{2}, \frac{w}{2}, 48)$	DECONV-(N64, K4×4, S2, P1), BN, LeakyReLU
output layer	$(\frac{h}{2}, \frac{w}{2}, 96) \rightarrow (h, w, 5)$	DECONV-(N5, K4×4, S2, P1), Tanh

List of Figures

List of Tables

References

[1] Gwenolé Quellec, Guy Cazuguel, Beatrice Cochener, and Mathieu Lamard, “Multiple-instance learning for medical image and video analysis,” *IEEE Reviews in Biomedical Engineering*, vol. 10, pp. 213–234, 2017. doi:10.1109/RBME.2017.2651164

[2] Constantinos Loukas, “Video content analysis of surgical procedures,” *Surgical Endoscopy*, vol. 32, pp. 553–568, 2017. doi:10.1007/s00464-017-5878-1

[3] S. Suganyadevi, V. Seethalakshmi, and K. Balasamy, “A review on deep learning in medical image analysis,” *International Journal of Multimedia Information Retrieval*, 2021. doi:10.1007/s13735-021-00218-1

[4] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F. Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, and Klaus H. Maier-Hein, “nnu-net: Self-adapting framework for u-net-based medical image segmentation,” *arXiv*, 2018. arXiv:1809.10486

[5] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, pp. 115–118, 2017. doi:10.1038/nature21056

[6] Stefanie Speidel, Julia Benzko, Sebastian Krappe, Gunther Sudra, Pedram Azad, Beat Peter Müller-Stich, Carsten Gutt, and Rüdiger Dillmann, “Automatic classification of minimally invasive instruments based on endoscopic image sequences,” in *Medical Imaging 2009: Visualization, Image-Guided Procedures, and Modeling*, vol. 7261, 2009, p. 72610A.

[7] Suren Kumar, Madusudanan Sathia Narayanan, Sukumar Misra, Sudha Garimella, Pankaj Singhal, Jason J. Corso, and Venkat Krovi, “Vision-based decision-support and safety systems for robotic surgery,” in *Proc. Medical Cyber Physical Systems*, 2013, p. 72610A.

[8] David Borgte, Florent Lalys, and Pierre Jannin, “Surgical tools recognition and pupil segmentation for cataract surgical process modeling,” *Stud Health Technol Inform*, vol. 173, pp. 78–84, 2012.

[9] Manfred Jürgen Primus, Klaus Schoeffmann, and Laszlo Böszörmenyi, “Instrument classification in laparoscopic videos,” in *13th international workshop on content-based multimedia indexing (CBMI)*, 2015, pp. 1–6. doi:10.1109/CBMI.2015.7153616

[10] Grant Haskins, Uwe Kruger, and Pingkun Yan, “Deep learning in medical image registration: a survey,” *Machine Vision and Applications*, vol. 31, p. 8, 2020. doi:10.1007/s00138-020-01060-x

[11] Fakhre Alam and Sami Ur Rahman, “Challenges and solutions in multimodal medical image subregion detection and registration,” *Journal of medical imaging and radiation sciences*, vol. 50, pp. 24–30, 2019. doi:10.1016/j.jmir.2018.06.001

[12] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy, “Deep learning techniques for medical image segmentation: Achievements and challenges,” *Journal of digital imaging*, vol. 32, pp. 582–596, 2019. doi:10.1007/s10278-019-00227-x

[13] Hiroshi Fujita, “AI-based computer-aided diagnosis (AI-CAD): the latest review to read first,” *Radiological Physics and Technology*, vol. 13, pp. 6–19, 2020. doi:10.1007/s12194-019-00552-4

[14] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, AnnetteKopp-Schneider, Bennett A. Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M. Summers, Bram van Ginneken, Michel Bilello, Patrick Bilic, Patrick F. Christ, Richard K. G. Do, Marc J. Gollub, Stephan H. Heckers, Henkjan Huisman, William R. Jarnagin, Maureen K. McHugo, Sandy Napel, Jennifer S. Goli Pernicka, Kawal Rhode, Catalina Tobon-Gomez, Eugene Vorontsov, Henkjan Huisman, James A. Meakin, Sebastien Ourselin, Manuel Wiesenfarth, Pablo Arbelaez, Byeonguk Bae, Sihong Chen, Laura Daza, Jianjiang Feng, Baochun He, Fabian Isensee, Yuanfeng Ji, Fucang Jia, Namkug Kim, Ildoo Kim, Dorit Merhof, Akshay Pai, Beomhee Park, Mathias Perslev, Ramin Rezaiifar, Oliver Rippel, Ignacio Sarasua, Wei Shen, Jaemin Son, Christian Wachinger, Liansheng Wang, Yan Wang, Yingda Xia, Daguang Xu, Zhanwei Xu, Yefeng Zheng, Amber L. Simpson, Lena Maier-Hein, and M. Jorge Cardoso, “The medical segmentation decathlon,” *arXiv*, 2021. arXiv:2106.05735

[15] Hassan Al Hajj, Mathieu Lamard, Pierre-Henri Conze, Béatrice Cochener, and Gwenolé Quellec, “Monitoring tool usage in surgery videos using boosted convolutional and recurrent neural networks,” *Medical image analysis*, vol. 47, pp. 203–218, 2018.

[16] Omri Bar, Daniel Neimark, Maya Zohar, Gregory D. Hager, Ross Girshick, Gerald M. Fried, Tamir Wolf, and Dotan Asselmann, “Impact of data on generalization of AI for surgical intelligence applications,” *Scientific Reports*, vol. 10, 2020. doi:10.1038/s41598-020-79173-6

[17] Nicolas Padoy, Tobias Blum, Seyed-Ahmad Ahmadi, Hubertus Feussner, Marie-Odile Berger, and Nassir Navab, “Statistical modeling and recognition of surgical workflow,” *Medical Image Analysis*, vol. 16, pp. 632–641, 2012. doi:10.1016/j.media.2010.10.001

[18] Shervin Minaee, Rahele Kafieh, Milan Sonka, Shakib Yazdani, and Ghazaleh Jamalipour Soufi, “Deep-COVID: Predicting COVID-19 from chest x-ray images using deep transfer learning,” *Medical Image Analysis*, vol. 65, p. 101794, 2020. doi:10.1016/j.media.2020.101794

[19] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3462–3471. doi:10.1109/CVPR.2017.369

[20] National Institutes of Health Clinical Center. Deeplession. Accessed: 01 Oct 2021. https://nihcc.app.box.com/v/DeepLesion

[21] Andru P. Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel de Mathelin, and Nicolas Padoy, “Endonet: A deep architecture for recognition tasks on laparoscopic videos,” *IEEE transactions on medical imaging*, vol. 36, pp. 86–97, 2016. doi:10.1109/TMI.2016.2593957

[22] Hassan Al Hajj, Mathieu Lamard, Pierre Henri Conze, Béatrice Cochener, and Gwenolé Quellec. (2021) CATARACTS. doi:10.21227/ac97-8m18

[23] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.

[24] Harsh Panwar, P. K. Gupta, Mohammad Khubeb Siddiqui, Ruben Morales-Menendez, and Vaishnavi Singh, “Application of deep learning for fast detection of COVID-19 in x-rays using nCOVnet,” *Chaos*, vol. 138, p. 109944, 2020. doi:10.1016/j.chaos.2020.109944

[25] Alexander Selvikvåg Lundervold and Arvid Lundervold, “An overview of deep learning in medical imaging focusing on MRI,” *Zeitschrift für Medizinische Physik*, vol. 29, pp. 102–127, 2019. doi:10.1016/j.zemedi.2018.11.002

[26] Jannik Schaaf, Martin Sedlmayr, Johanna Schaefer, and Holger Storf, “Diagnosis of rare diseases: a scoping review of clinical decision support systems,” *Orphanet Journal of Rare Diseases*, vol. 15, 2020. doi:10.1186/s13023-020-01536-z

[27] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan, “GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification,” *Neurocomputing*, vol. 321, pp. 321–331, 2018. doi:https://doi.org/10.1016/j.neucom.2018.09.013

[28] Yuya Onishi, Atsushi Teramoto, Masakazu Tsujimoto, Tetsuya Tsukamoto, Kuniaki Saito, Hiroshi Toyama, Kazuyoshi Imaizumi, and Hiroshi Fujita, “Automated pulmonary nodule classification in computed tomography images using a deep convolutional neural network trained by generative adversarial networks,” *BioMed Research International*, vol. 2019, p. 6051939, 2019. doi:10.1155/2019/6051939

[29] Haoxiang Wang and S. Smys, “Overview of configuring adaptive activation functions for deep neural networks - a comparative study,” *Journal of Ubiquitous Computing and Communication Technologies*, vol. 3, pp. 10–22, 2021. doi:10.36548/jucct.2021.1.002

[30] Simon S. Haykin, *Neural networks and learning machines*, vol. 10. Upper Saddle River, NJ: Pearson, 2009.

[31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–44, 2015. doi:10.1038/nature14539

[32] Hochreiter Sepp, “Untersuchungen zu dynamischen neuronalen Netzen,” PhD thesis, Technische Universität München, 1991.

[33] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, 1997. doi:10.1162/neco.1997.9.8.1735

[34] Long_Short_Term_Memory.png. Accessed: 28 Jan 2022, licensed under the Creative Commons Attribution-Share Alike 4.0 International license. https://commons.wikimedia.org/wiki/File:Long_Short_Term_Memory.png

[35] Felix A. Gers and Jürgen Schmidhuber, “Recurrent nets that time and count,” in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 3. IEEE, 2000, pp. 189–194. doi:10.1109/IJCNN.2000.861302

[36] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[37] Lars M. Mescheder, Andreas Geiger, and Sebastian Nowozin, “Which training methods for GANs do actually converge?” in *ICML*, 2018.

[38] Martin Arjovsky, Soumith Chintala, and Léon Bottou, “Wasserstein generative adversarial networks,” in *International conference on machine learning.* PMLR, 2017, pp. 214–223.

[39] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas, “A metric for distributions with applications to image databases,” in *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, 1998, pp. 59–66.

[40] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[41] Choi Yunjey, Choi Min-Je, Kim Mun Su, Ha Jung-Woo, Kim Sunghun, and Choo Jaegul, “StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[42] Goodfellow Ian, Bengio Yoshua, and Courville Aaron, *Deep learning.* MIT Press, 2016.

[43] Jorge Nocedal and Stephen J. Wright, *Numerical optimization*, Springer Series in Operations Research and Financial Engineering. New York, NY: Springer New York, 2006. doi:10.1007/978-0-387-40065-5

[44] Bottou Léon, “On-line learning and stochastic approximations,” in *In On-line Learning in Neural Networks.* Cambridge University Press, 1998, pp. 9–42.

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

[46] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun, “Adaptive gradient methods with dynamic bound of learning rate,” *arXiv*, 2019. arXiv:1902.09843

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015. doi:10.1007/s11263-015-0816-y

[48] Sameer Trikha, Andrew Michael John Turnbull, R. J. Morris, David F. Anderson, and Parwez Hossain, “The journey to femtosecond laser-assisted cataract surgery: new beginnings or a false dawn?” *Eye*, vol. 27, pp. 461–473, 2013. doi:10.1038/eye.2012.293

[49] Augenärzte Zürich. Wie entsteht der Graue Star (Katarakt)? Accessed: 01 Oct 2021. https://www.augenaerzte-wallisellen.ch/wie-entsteht-der-graue-star/

[50] Cataracts.png. Accessed: 28 Jan 2022, licensed under the Creative Commons Attribution-Share Alike 4.0 International license. https://commons.wikimedia.org/wiki/File:Cataracts.png

[51] H. Burkhard Dick, Ronald D. Gerste, and Tim Schultz, *Femtosecond laser surgery in ophthalmology.* Thieme, 2018. doi:10.1055/b-0038-149386

[52] Eric J. Linebarger, David R. Hardten, Gaurav K. Shah, and Richard L. Lindstrom, “Phacoemulsification and modern cataract surgery,” *Survey of Ophthalmology*, vol. 44, pp. 123–147, 1999. doi:10.1016/S0039-6257(99)00085-5

[53] Amit Dr. Sood. Laparoscopic cholecystectomy. Accessed: 01 Oct 2021. https://dramitsood.com/laparoscopic-cholecystectomy/

[54] Aslam Ejaz, Gaya Spolverato, Yuhree Kim, Rebecca Dodson, Jason K. Sicklick, Henry A. Pitt, Keith D. Lillemoe, John L. Cameron, and Timothy M. Pawlik, “Long-term health-related quality of life after iatrogenic bile duct injury repair,” *Journal of the American College of Surgeons*, vol. 219, pp. 923–932.e10, 2014. doi:10.1016/j.jamcollsurg.2014.04.024

[55] Jeffrey P. Okeson, *Management of temporomandibular disorders and occlusion-E-book*. Elsevier Health Sciences, 2019.

[56] Nils Nicolay, Frank Antwerpen, Bijan Fink, and Julian Wütscher. Dysgnathie. Accessed: 01 Oct 2021. https://flexikon.doccheck.com/de/Dysgnathie

[57] Mark A. Hurt, "Weedon d. weedon's skin pathology. 3rd ed. london: Churchill livingstone elsevier, 2010," *Dermatology Practical & Conceptual*, vol. 2, 2012. doi:10.5826/dpc.0201a15

[58] Paul Ridder, *Craniomandibuläre Dysfunktion: Interdisziplinäre Diagnose- und Behandlungsstrategien*. Elsevier Health Sciences, 2019.

[59] Uniklinik RWTH Aachen. Dysgnathie-chirurgie. Accessed: 01 Oct 2021. https://www.ukaachen.de/kliniken-institute/klinik-fuer-mund-kiefer-und-gesichtschirurgie/fuer-patienten/behandlungsspektrum/dysgnathie-chirurgie/

[60] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A. W. M. van der Laak, Bram van Ginneken, and Clara I. Sánchez, "A survey on deep learning in medical image analysis," *Med Image Anal. (2017) 42:60-88*, 2017. doi:10.1016/j.media.2017.07.005

[61] 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, and Gwenolé Quellec, "CATARACTS: Challenge on automatic tool annotation for cataRACT surgery," *Medical image analysis*, vol. 52, pp. 24–41, 2019. doi:10.1016/j.media.2018.11.008

[62] Andru P. Twinanda, Didier Mutter, Jacques Marescaux, Michel de Mathelin, and Nicolas Padoy, "Single-and multi-task architectures for tool presence detection challenge at m2cai 2016," *arXiv*, 2016. arXiv:1610.08851

[63] Manish Sahu, Anirban Mukhopadhyay, Angelika Szengel, and Stefan Zachow, "Tool and phase recognition using contextual CNN features," *arXiv*, 2016. arXiv:1610.08854

[64] Ashwin Raju, Sheng Wang, and Junzhou Huang, "M2CAI surgical tool detection challenge report," in *Workshop and Challenges on Modeling and Monitoring of Computer Assisted Intervention (M2CAI), Athens, Greece, Technical report*, 2016.

[65] Igor Pernek and Alois Ferscha, "A survey of context recognition in surgery," *Medical & biological engineering & computing*, vol. 55, pp. 1719–1734, 2017. doi:10.1007/s11517-017-1670-6

[66] Constantinos Loukas, "Video content analysis of surgical procedures," *Surgical endoscopy*, vol. 32, pp. 553–568, 2018. doi:10.1007/s00464-017-5878-1

[67] Shoji Morita, Hitoshi Tabuchi, Hiroki Masumoto, Tomofusa Yamauchi, and Naotake Kamiura, "Real-time extraction of important surgical phases in cataract surgery videos," *Scientific reports*, vol. 9, p. 16590, 2019. doi:10.1038/s41598-019-53091-8

[68] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998. doi:10.1109/5.726791

[69] Alex Krizhevsky, "Learning multiple layers of features from tiny images," *Citeseer*, 2009.

[70] Karen Simonyan and Zisserman Andrew, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. doi:10.1109/CVPR.2016.90

[72] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269. doi:10.1109/CVPR.2017.243

[73] Christian Szegedy, We Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9. doi:10.1109/CVPR.2015.7298594

[74] Paritosh Parmar and Brendan Tran Morris, "Learning to score olympic events," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 20–28.

[75] Taeoh Kim, Hyeongmin Lee, MyeongAh Cho, Ho Seong Lee, Dong Heon Cho, and Sangyoun Lee, "Learning temporally invariant and localizable features via data augmentation for video recognition," in *Computer Vision – ECCV 2020 Workshops*, Lecture Notes in Computer Science, vol. 12536, 2020, pp. 386–403. doi:10.1007/978-3-030-66096-3_27

[76] Jingwei Ji, Kaidi Cao, and Juan Carlos Niebles, "Learning temporal action proposals with fewer labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[77] Simon Niklaus, Long Mai, and Feng Liu, "Video frame interpolation via adaptive convolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. arXiv:1708.01692

[78] Felix Yu, Gianluca Silva Croso, Tae Soo Kim, Ziang Song, Felix Parker, Gregory D. Hager, Austin Reiter, Swaroop S. Vedula, Haider Ali, and Shameema Sikder, "Assessment of automated identification of phases in videos of cataract surgery using machine learning and deep learning techniques," *JAMA network open*, vol. 2, p. e191860, 2019. doi:10.1001/jamanetworkopen.2019.1860

[79] Odysseas Zisimopoulos, Evangello Flouty, Imanol Luengo, Petros Giataganas, Jean Nehme, Andre Chow, and Danail Stoyanov, "Deepphase: Surgical phase recognition in CATARACTS videos," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Lecture Notes in Computer Science, vol. 11073, 2018, pp. 265–272. doi:10.1007/978-3-030-00937-3_31

[80] Fabian Isensee, Paul Jäger, Jakob Wasserthal, David Zimmerer, Jens Petersen, Simon Kohl, Justus Schock, Andre Klein, Tobias Roß, Sebastian Wirkert, Peter Neher, Stefan Dinkelacker, Gregor Köhler, and Klaus Maier-Hein. (2020, 1) batchgenerators - a python framework for data augmentation. doi:10.5281/zenodo.3632567

[81] Dinggang Shen, Guorong Wu, and Heung-Il Suk, "Deep learning in medical image analysis," *Annual review of biomedical engineering*, vol. 19, pp. 221–248, 2017. doi: 10.1146/annurev-bioeng-071516-044442

[82] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2015.

[83] Paula Branco, Luís Torgo, and Rita P. Ribeiro, "Relevance-based evaluation metrics for multi-class imbalanced domains," in *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, vol. 10234, 2017, pp. 698–710. doi:10.1007/978-3-319-57454-7_54

[84] Rasiah Bharathan, Rajesh Aggarwal, and Ara Darzi, "Operating room of the future," *Best Practice & Research Clinical Obstetrics & Gynaecology*, vol. 27, pp. 311–322, 2013. doi:10.1016/j.bpobgyn.2012.11.003

[85] Katia Charriere, Gwenolé Quellec, Mathieu Lamard, Gouenou Coatrieux, Beatrice Cochener, and Guy Cazuguel, "Automated surgical step recognition in normalized cataract surgery videos," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 4647–4650. doi:10.1109/EMBC.2014.6944660

[86] Lena Maier-Hein, Swaroop S. Vedula, Stefanie Speidel, Nassir Navab, Ron Kikinis, Adrian Park, Matthias Eisenmann, Hubertus Feussner, Germain Forestier, Stamatia Giannarou, Makoto Hashizume, Darko Katic, Hannes Kenngott, Michael Kranzfelder, Anand Malpani, Keno März, Thomas Neumuth, Nicolas Padoy, Carla Pugh, Nicolai Schoch, Danail Stoyanov, Russell Taylor, Martin Wagner, Gregory D. Hager, and Pierre Jannin, "Surgical data science for next-generation interventions," *Nature Biomedical Engineering*, vol. 1, pp. 691–696, 2017. doi:10.1038/s41551-017-0132-7

[87] Nicolas Padoy, "Machine and deep learning for workflow recognition during surgery," *Minimally Invasive Therapy & Allied Technologies*, vol. 28, pp. 82–90, 2019. doi:10.1080/13645706.2019.1584116

[88] Nathalie Bricon-Souf and Conrad R. Newman, "Context awareness in health care: A review," *International Journal of Medical Informatics*, vol. 76, pp. 2–12, 2007. doi:10.1016/j.ijmedinf.2006.01.003

[89] Olga Dergachyova, David Bouget, Arnaud Huaulmé, Xavier Morandi, and Pierre Jannin, "Automatic data-driven real-time segmentation and recognition of surgical workflow," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, pp. 1081–1089, 2016. doi:10.1007/s11548-016-1371-x

[90] Gwenolé Quellec, Mathieu Lamard, Beatrice Cochener, and Guy Cazuguel, "Real-time task recognition in cataract surgery videos using adaptive spatiotemporal polynomials," *IEEE Transactions on Medical Imaging*, vol. 34, pp. 877–887, 2015. doi:10.1109/TMI.2014.2366726

[91] Arnaud Huaulmé, Pierre Jannin, Fabian Reche, Jean-Luc Faucheron, Alexandre Moreau-Gaudry, and Sandrine Voros, "Offline identification of surgical deviations in laparoscopic rectopexy," *Artificial Intelligence in Medicine*, vol. 104, p. 101837, 2020. doi:10.1016/j.artmed.2020.101837

[92] Dazzi Luisella, Fassino Clara, Saracco Roberta, Quaglini Silvana, and Stefanelli Mario, "A patient workflow management system built on guidelines," in *AMIA 1997, American Medical Informatics Association Annual Symposium, Nashville, TN, USA, October 25-29, 1997*. AMIA, 1997.

[93] Beenish Bhatia, Tim Oates, Yan Xiao, and Peter Hu, "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. AAAI Press, 2007, p. 1761–1766. doi:10.5555/1620113.1620126

[94] Germain Forestier, Laurent Riffaud, and Pierre Jannin, "Automatic phase prediction from low-level surgical activities," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, pp. 833–841, 2015. doi:10.1007/s11548-015-1195-0

[95] Tobias Blum, Hubertus Feußner, and Nassir Navab, “Modeling and segmentation of surgical workflow from laparoscopic video,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, Lecture Notes in Computer Science, vol. 6363, 2010, pp. 400–407. doi:10.1007/978-3-642-15711-0_50

[96] Yueming Jin, Yonghao Long, Cheng Chen, Zixu Zhao, Qi Dou, and Pheng-Ann Heng, “Temporal memory relation network for workflow recognition from surgical video,” *IEEE Transactions on Medical Imaging*, 2021.

[97] Jakob E. Bardram, Afsaneh Doryab, Rune M. Jensen, Poul M. Lange, Kristian L. G. Nielsen, and Soren T. Petersen, “Phase recognition during surgical procedures using embedded and body-worn sensors,” in *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2011, pp. 45–53. doi:10.1109/PERCOM.2011.5767594

[98] Matthew Stephen Holden, Tamas Ungi, Derek Sargent, Robert C. McGraw, Elvis C. S. Chen, Sugantha Ganapathy, Terry M. Peters, and Gabor Fichtinger, “Feasibility of real-time workflow segmentation for tracked needle interventions,” *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 1720–1728, 2014. doi:10.1109/TBME.2014.2301635

[99] Seyed-Ahmad Ahmadi, Tobias Sielhorst, Ralf Stauder, Martin Horn, Hubertus Feussner, and Nassir Navab, “Recovery of surgical workflow without explicit models,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, Lecture Notes in Computer Science, vol. 4190, Berlin, Germany, 2006, pp. 420–428. doi:10.1007/11866565_52

[100] Niclas Padoy, Tobias Blum, Irfan Essa, Hubertus Feussner, Marie-Odile Berger, and Nassir Navab, “A boosted segmentation method for surgical workflow analysis,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, Lecture Notes in Computer Science, vol. 4791, 2007, pp. 102–109. doi:10.1007/978-3-540-75757-3_13

[101] Tobias Blum, Nicolas Padoy, Hubertus Feußner, and Nassir Navab, “Modeling and online recognition of surgical phases using hidden markov models,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008*, Lecture Notes in Computer Science, vol. 5242, 2008, pp. 627–635. doi:10.1007/978-3-540-85990-1_75

[102] Nicolas Padoy, Tobias Blum, Hubertus Feussner, Marie-Odile Berger, and Nassir Navab, “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. AAAI Press, 2008, p. 1718–1724. doi:10.5555/1620138.1620155

[103] Loubna Bouarfa, Pieter P. Jonker, and Jenny Dankelman, “Surgical context discovery by monitoring low-level activities in the OR,” in *MICCAI workshop on modeling and monitoring of computer assisted interventions (M2CAI). London, UK*, 2009.

[104] Ralf Stauder, Asli Okur, Loïc Peter, Armin Schneider, Michael Kranzfelder, Hubertus Feussner, and Nassir Navab, “Random forests for phase detection in surgical workflow analysis,” in *Information Processing in Computer-Assisted Interventions*, Lecture Notes in Computer Science, vol. 8498, 2014, pp. 148–157. doi:10.1007/978-3-319-07521-1_16

[105] Robert DiPietro, Ralf Stauder, Ergün Kayis, Armin Schneider, Michael Kranzfelder, Hubertus Feussner, Gregory D. Hager, and Nassir Navab, “Automated surgical-phase recognition using rapidly-deployable sensors,” in *Proc MICCAI Workshop M2CAI*, 2015.

[106] Rémi Cadène, Thomas Robert, Nicolas Thome, and Matthieu Cord, “M2CAI workflow challenge: Convolutional neural networks with time smoothing and hidden markov model for video frames classification,” *arXiv*, 2016. arXiv:1610.05541

[107] Colin Lea, Joon Hyuck Choi, Austin Reiter, and Gregory Hager, "Surgical phase recognition: from instrumented ORs to hospitals around the world," in *Medical image computing and computer-assisted intervention M2CAI'97MICCAI workshop*, 2016, pp. 45–54.

[108] Isabel Funke, Alexander Jenke, Sören Torge Mees, Jürgen Weitz, Stefanie Speidel, and Sebastian Bodenstedt, "Temporal coherence-based self-supervised learning for laparoscopic workflow analysis," in *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, Lecture Notes in Computer Science, vol. 11041, 2018, pp. 85–93. doi:10.1007/978-3-030-01201-4_11

[109] Yueming Jin, Qi Dou, Hao Chen, Lequan Yu, Jing Qin, Chi-Wing Fu, and Pheng-Ann Heng, "SV-RCNet: Workflow recognition from surgical videos using recurrent convolutional network," *IEEE Transactions on Medical Imaging*, vol. 37, pp. 1114–1126, 2018. doi:10.1109/TMI.2017.2787657

[110] Constantinos Loukas, "Surgical phase recognition of short video shots based on temporal modeling of deep features," in *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies*, 2019, pp. 21–29. doi:10.5220/0007352000210029

[111] Babak Namazi, Ganesh Sankaranarayanan, and Venkat Devarajan, "Automatic detection of surgical phases in laparoscopic videos," in *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, 2018, pp. 124–130.

[112] Gaurav Yengera, Didier Mutter, Jacques Marescaux, and Nicolas Padoy, "Less is more: Surgical phase recognition with less annotations through self-supervised pre-training of CNN-LSTM networks," *arXiv*, 2018. arXiv:1805.08569

[113] Daichi Kitaguchi, Nobuyoshi Takeshita, Hiroki Matsuzaki, Hiroaki Takano, Yohei Owada, Tsuyoshi Enomoto, Tatsuya Oda, Hirohisa Miura, Takahiro Yamanashi, Masahiko Watanabe, Daisuke Sato, Yusuke Sugomori, Seigo Hara, and Masaaki Ito, "Real-time automatic surgical phase recognition in laparoscopic sigmoidectomy using the convolutional neural network-based deep learning approach," *Surgical Endoscopy*, vol. 34, pp. 4924–4931, 2019. doi:10.1007/s00464-019-07281-0

[114] Yueming Jin, Huaxia Li, Qi Dou, Hao Chen, Jing Qin, Chi-Wing Fu, and Pheng-Ann Heng, "Multi-task recurrent convolutional network with correlation loss for surgical video analysis," *Medical Image Analysis*, vol. 59, p. 101572, 2020. doi:10.1016/j.media.2019.101572

[115] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[116] M2cai16-workflow. Accessed: 01 Oct 2021. http://camma.u-strasbg.fr/result-list-m2cai16-workflow

[117] Fangqiu Yi and Tingting Jiang, "Not end-to-end: Explore multi-stage architecture for online surgical phase recognition," *arXiv*, 2021. arXiv:2107.04810

[118] Gwen R. J. Swennen, *3d virtual treatment planning of orthognathic surgery.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 217–277. doi:10.1007/978-3-662-47389-4_3

[119] Alicia Hernández Salazar, Juan A. Juanes Méndez, and Francisco Pastor Vázquez, "DOLPHIN 3D," *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM '14*, pp. 35–40, 2014. doi:10.1145/2669711.2669875

[120] M Meehan, M Teschner, and Sabine Girod, "Three-dimensional simulation and prediction of craniofacial surgery," *Orthodontics & craniofacial research*, vol. 6, pp. 102–107, 2003.

[121] Anders Westermark, Stefan Zachow, and Barry L. Eppley, "Three-dimensional osteotomy planning in maxillofacial surgery including soft tissue prediction," *Journal of Craniofacial Surgery*, vol. 16, pp. 100–104, 2005.

[122] Paul G.M. Knoops, Alessandro Borghi, Federica Ruggiero, Giovanni Badiali, Alberto Bianchi, Claudio Marchetti, Naiara Rodriguez-Florez, Richard W.F. Breakey, Owase Jeelani, David J. Dunaway, et al., "A novel soft tissue prediction methodology for orthognathic surgery based on probabilistic finite element modelling," *PloS one*, vol. 13, p. e0197209, 2018.

[123] Wouter Mollemans, Filip Schutyser, Nasser Nadjmi, Frederik Maes, and Paul Suetens, "Predicting soft tissue deformations for a maxillofacial surgery planning system: from computational strategies to a complete clinical validation," *Medical image analysis*, vol. 11, pp. 282–301, 2007.

[124] Lucia H.C. Cevidanes, Scott Tucker, Martin Styner, Hyungmin Kim, Jonas Chapuis, Mauricio Reyes, William Proffit, Timothy Turvey, and Michael Jaskolka, "Three-dimensional surgical simulation," *American journal of orthodontics and dentofacial orthopedics*, vol. 138, pp. 361–371, 2010.

[125] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2223–2232.

[126] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z. Li, "Face alignment in full pose range: A 3d total solution," *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[127] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8789–8797.

[128] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha, "StarGAN v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8188–8197.

[129] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua, "Towards open-set identity preserving face synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6713–6722.

[130] Shuang Liu, Dan Li, Tianchi Cao, Yuke Sun, Yingsong Hu, and Junwen Ji, "GAN-based face attribute editing," *IEEE Access*, vol. 8, pp. 34 854–34 867, 2020.

[131] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen, "Attgan: Facial attribute editing by only changing what you want," *IEEE Transactions on Image Processing*, vol. 28, pp. 5464–5478, 2019.

[132] Evgeny Gladilin and Alexander Ivanov, "Computational modelling and optimisation of soft tissue outcome in cranio-maxillofacial surgery planning," *Computer methods in biomechanics and biomedical engineering*, vol. 12, pp. 305–318, 2009.

[133] Bryan T Harris, Daniel Montero, Gerald T. Grant, Dean Morton, Daniel R. Llop, and Wei-Shao Lin, "Creation of a 3-dimensional virtual dental patient for computer-guided surgery and CAD-CAM interim complete removable and fixed dental prostheses: A clinical report," *The Journal of prosthetic dentistry*, vol. 117, pp. 197–204, 2017. doi:10.1016/j.prosdent.2016.06.012

[134] Josep Rubio-Palau, Alejandra Prieto-Gundin, Asteria Albert Cazalla, Miguel Bejarano Serrano, Gemma Garcia Fructuoso, Francisco Parri Ferrandis, and Alejandro Rivera Baró,

"Three-dimensional planning in craniomaxillofacial surgery," *Annals of maxillofacial surgery*, vol. 6, pp. 281–286, 2016. doi:10.4103/2231-0746.200322

[135] Pratik Premjani, Anas Hasan Al-Mulla, and Donald J. Ferguson, "Accuracy of 3d facial models obtained from CBCT volume wrapping," *Journal of clinical orthodontics : JCO*, vol. 49, pp. 641–6, 2015.

[136] Christopher Lane and William Harrell, "Completing the 3-dimensional picture," *American journal of orthodontics and dentofacial orthopedics: official publication of the American Association of Orthodontists, its constituent societies, and the American Board of Orthodontics*, vol. 133, pp. 612–20, 2008. doi:10.1016/j.ajodo.2007.03.023

[137] Shaobo Guan. (2018) TL-GAN: transparent latent-space GAN. Accessed: 2020-07-13. https://github.com/SummitKwan/transparent_latent_gan/

[138] Wei Shen and Rujie Liu, "Learning residual images for face attribute manipulation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4030–4038.

[139] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh, "Recycle-GAN: Unsupervised video retargeting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 119–135.

[140] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter, "A 3d face model for pose and illumination invariant face recognition," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2009, pp. 296–301.

[141] Aaron Gokaslan, Vivek Ramanujan, Daniel Ritchie, Kwang In Kim, and James Tompkin, "Improving shape deformation in unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 649–665.

[142] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595.

[143] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou, "3d face morphable models "in-the-wild"," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5464–5473.

[144] Patrik Huber, Guosheng Hu, Rafael Tena, Pouria Mortazavian, Koppen P. Willem, William Christmas, Matthias Rätsch, and Josef Kittler, "A multiresolution 3d morphable face model and fitting framework," in *Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2016.

[145] Aaron S. Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos, "Large pose 3d face reconstruction from a single image via direct volumetric CNN regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1031–1039.

[146] Matan Sela, Elad Richardson, and Ron Kimmel, "Unrestricted facial geometry reconstruction using image-to-image translation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1576–1585.

[147] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni, "Regressing robust and discriminative 3d morphable models with a very deep neural network," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[148] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou, “Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1155–1164.

[149] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville, “Improved training of wasserstein GANs,” in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[150] Diederik P. Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2015.

[151] Ariadna Quattoni and Antonio Torralba, “Recognizing indoor scenes,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 413–420.

[152] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio, “The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 11–19.

[153] Adrian Bulat and Georgios Tzimiropoulos, “How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks),” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1021–1030.

[154] Yao Feng. (2018) Face3d: Python tools for processing 3d face. https://github.com/YadiraF/face3d

[155] Hang Dai, Nick Pears, William Smith, and Christian Duncan, “Statistical modeling of craniofacial shape and texture,” *International Journal of Computer Vision*, vol. 128, pp. 547–571, 2019. doi:10.1007/s11263-019-01260-7

[156] Tim Esler. Pretrained pytorch face detection (MTCNN) and recognition (inceptionresnet) models. https://github.com/timesler/facenet-pytorch

[157] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman, “Vggface2: A dataset for recognising faces across pose and age,” in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, 2018, pp. 67–74.

[158] Zhou Wang, Alan Conrad. Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004. doi:10.1109/TIP.2003.819861

List of Publications and Supervised Thesis

Journal Articles

- Matthias Schaufelberger, Reinald Peter Kuehle, Frederic Weichel, Niclas Hagen, **Andreas Wachter**, Friedemann Ringwald, Urs Eisenmann, Christian Freudlsperger, Michael Engel, and Werner Nahm. *A statistical shape model for radiation-free assessment and classification ofcraniosynostosis.* Elsevier Medical Image Analysis, Oct. 2021., under review.
- Tobias Gerach, Steffen Schuler, **Andreas Wachter**, and Axel Loewe. *The impact of standard strategies for ablation of atrialfibrillation on cardiovascular performance in a four-chamberheart model.* Biomech Model Mechanobiol, Oct. 2021., under review.
- **Andreas Wachter**, and Werner Nahm. *Workflow Augmentation of Video Data for Event Recognition with Time-Sensitive Neural Networks.* preprint arXiv:2109.15063.
- Andlauer Robin*, **Andreas Wachter***, Matthias Schaufelberger, Frederic Weichel, Reinald Kuhle, Christian Freudlsperger, and Werner Nahm. *3D-Guided Face Manipulation of 2D Images for the Prediction of Post-Operative Outcome After Cranio-Maxillofacial Surgery.* IEEE Transactions on Image Processing vol. 30, pp. 7349–7363, Jan. 2021, doi:10.1109/TIP.2021.3096081.
- **Andreas Wachter**, Jan Kost, and Werner Nahm. *Simulation-Based Estimation of the Number of Cameras Required for 3D Reconstruction in a Narrow-Baseline Multi-Camera Setup.* Journal of Imaging vol. 7, no. 5, p. 87, Jan. 2021, doi:10.3390/jimaging7050087.
- Steffen Schuler, **Andreas Wachter**, and Olaf Dössel. *Electrocardiographic Imaging Using a Spatio-Temporal Basis of Body Surface Potentials—Application to Atrial Ectopic Activity.* Frontiers in Physiology vol. 9:1126, Aug. 2018. doi:10.3389/fphys.2018.01126.

Refereed Conference Articles

- Matthias Schaufelberger, Reinald Peter Kuehle, Frederic Weichel, **Andreas Wachter**, Niclas Hagen, Friedemann Ringwald, Urs Eisenmann, Christian Freudlsperger, and Werner Nahm. *Laplace-Beltrami Refined Shape Regression Applied* *to Neck Recon- struction for Craniosynostosis Patients.* In Current Directions in Biomedical Engineering, De Gruyter, 2021, submitted.

- **Andreas Wachter**, Adrian Mohra, and Werner Nahm. *Development of a Real-Time Virtual Reality Environment for Visualization of Fully Digital Microscope Datasets.* In Proceedings of SPIE, 10868, Advanced Biomedical and Clinical Diagnostic and Surgical Guidance Systems XVII, 108681F, San Francisco, California, United States: SPIE BiOS, 2019. doi:10.1117/12.2506898.
- **Andreas Wachter**, Jan Kost, and Werner Nahm. *MATLAB Simulation Environment for Estimating the Minimal Number and Positions of Cameras for 3D Surface Reconstruction in a Fully-Digital Surgical Microscope.* In Current Directions in Biomedical Engineering, De Gruyter, 2018, vol. 4, no. 1, pp. 517–520. doi:10.1515/cdbme-2018-0124.
- Elisa Dicke, **Andreas Wachter**, and Werner Nahm. *Estimation of the Interpolation Error of a Three-Step Rotation Algorithm Using Recorded Images with Rotated Test Pattern as Ground Truth.* In Current Directions in Biomedical Engineering, De Gruyter, 2017, vol. 3, no. 2, pp. 555–558. doi:10.1515/cdbme-2017-0186.
- Christian Marzi, **Andreas Wachter**, and Werner Nahm. *Design of an Experimental Four-Camera Setup for Enhanced 3D Surface Reconstruction in Microsurgery.* In Current Directions in Biomedical Engineering, De Gruyter, 2017, vol. 3, no. 2, pp. 539-542. doi:10.1515/cdbme-2017-0185.
- **Andreas Wachter**, Axel Loewe, Martin W Krueger, Olaf Dössel, and Gunnar Seemann. *Mesh Structure-Independent Modeling of Patient-Specific Atrial Fiber Orientation.* Current Directions in Biomedical Engineering, De Gruyter, 2015, vol. 1, no. 1, pp. 409–412. doi:10.1515/cdbme-2015-0099.

Refereed Conference Abstracts

- **Andreas Wachter**, Robin Andlauer, and Werner Nahm. *Investigation of the Potential of CycleGANs for Generating Artificial Photorealistic Images.* In BMT 2020. 54h Annual Conference of the German Society for Biomedical Engineering. vol. 65, no. s1, 2020, pp. 267-273. doi:/10.1515/bmt-2020-6046.
- **Andreas Wachter**, and Werner Nahm. *Qualitative Analysis of Feature Extraction on Medical Images of a pre-trained Neural Network.* In 41st International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin: 2019.
- **Andreas Wachter** and Werner Nahm. *Workflow Model Generation for Video Augmentation for Neural Networks.* In BMT 2017. 53h Annual Conference of the German Society for Biomedical Engineering. Frankfurt: VDE / DGBMT, 2019. doi:/10.1515/bmt-2019-60032.
- **Andreas Wachter** and Werner Nahm. *Requirements for a Fully-Digital Surgical Microscope regarding the State of the Art of Surgical Microscopes and the Surgeon's*

Visual Perception. In BMT 2017. 51h Annual Conference of the German Society for Biomedical Engineering. Dresden: VDE / DGBMT, 2017. doi:/10.1515/bmt-2017-5019.

Conference Presentations

- **Andreas Wachter**, Robin Andlauer, and Werner Nahm. *Investigation of the Potential of CycleGANs for Generating Artificial Photorealistic Images.* In BMT 2020. 54h Annual Conference of the German Society for Biomedical Engineering. vol. 65, no. s1, 2020, pp. 267-273. doi:/10.1515/bmt-2020-6046.
- **Andreas Wachter** and Werner Nahm. *Workflow Model Generation for Video Augmentation for Neural Networks.* In BMT 2017. 53h Annual Conference of the German Society for Biomedical Engineering. Frankfurt: VDE / DGBMT, 2019. doi:/10.1515/bmt-2019-6003.
- **Andreas Wachter**, and Werner Nahm. *Qualitative Analysis of Feature Extraction on Medical Images of a pre-trained Neural Network.* In 41st International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin: 2019.
- **Andreas Wachter**, Adrian Mohra, and Werner Nahm. *Development of a Real-Time Virtual Reality Environment for Visualization of Fully Digital Microscope Datasets.* In Proceedings of SPIE, 10868, Advanced Biomedical and Clinical Diagnostic and Surgical Guidance Systems XVII, 108681F, San Francisco, California, United States: SPIE BiOS, 2019. doi:10.1117/12.2506898.
- **Andreas Wachter**, and Werner Nahm. *MICCAI 2018 - ENDOSCOPIC VISION CHALLENGE: CATARACTS -Team IBT-.* 21th International Conference on Medical Image Computing and Computer Assisted Intervention Granada, Spain: MICCAI 2018.
- **Andreas Wachter**, Jan Kost, and Werner Nahm. *MATLAB Simulation Environment for Estimating the Minimal Number and Positions of Cameras for 3D Surface Reconstruction in a Fully-Digital Surgical Microscope.* In Current Directions in Biomedical Engineering, De Gruyter, 2018, vol. 4, no. 1, pp. 517–520. doi:10.1515/cdbme-2018-0124.
- Elisa Dicke, **Andreas Wachter**, and Werner Nahm. *Estimation of the Interpolation Error of a Three-Step Rotation Algorithm Using Recorded Images with Rotated Test Pattern as Ground Truth.* In Current Directions in Biomedical Engineering, De Gruyter, 2017, vol. 3, no. 2, pp. 555–558. doi:10.1515/cdbme-2017-0186.
- **Andreas Wachter** and Werner Nahm. *Requirements for a Fully-Digital Surgical Microscope regarding the State of the Art of Surgical Microscopes and the Surgeon's Visual Perception.* In BMT 2017. 51h Annual Conference of the German Society for Biomedical Engineering. Dresden: VDE / DGBMT, 2017. doi:/10.1515/bmt-2017-5019.

- **Andreas Wachter**, Axel Loewe, Martin W Krueger, Olaf Dössel, and Gunnar Seemann. *Mesh Structure-Independent Modeling of Patient-Specific Atrial Fiber Orientation.* Current Directions in Biomedical Engineering, De Gruyter, 2015, vol. 1, no. 1, pp. 409–412. doi:10.1515/cdbme-2015-0099.

Reports and Thesis

- **Andreas Wachter**. *Model-Structure-Independent, Rule-Based Annotation of Atrial Fiber Orientation and Ablation Patterns for an in Silico Assessment of their Arrhythmogenic Potential*, Diploma thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2015.

Supervised Students

- Simon Kretz. *Atomatische Bestimmung der extrinsischen Kameraparameter eines Multi-Kameraaufbaus mit Hauptobjektiv*, Bachelor's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2020.
- Robin Andlauer. *Bidirectional translation between facial image and 3D reconstruction for postoperative face prediction*, Master's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2020.
- Jan Kost. *Erweiterung des Kamerasimulations-Frameworks zur Bestimmung der minimal benötigten Anzahl an Kameras für die 3D Rekonstruktion*, Research Projects, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2019 - 2020.
- Andreas Troschke. *Entwicklung eines Konzeptes zum Speichern gleichzeitiger Aufnahmen eines Multikameraaufbaus*, Bachelor's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2019.
- Kristina Geistert. *Analyse der Kataraktoperation aus entscheidungstheoretischer Sicht*, Research Projects, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2018 - 2019.
- Thomas Trapp. *Conceptualizing the Automatic Generation of Synthetic Training Data for LSTMs For Tool Recognition in Cataract Surgery Videos*, Bachelor's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2019.
- Jan Kost. *Estimation of the Minimal Number and Positions of Cameras for 3D Reconstruction Based on a 3D Reference Object Regarding the Anatomical Structure of Typical Microsurgery Scenes*, Bachelor's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2018.

- Adrian Mohra. *Umsetzung eines zweiten Benutzers in der selben VR-Reality*, Research Projects, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2018.
- Adrian Mohra. *Konzeptionierung einer Darstellung von medizinischen mikroskopischen 3D Datensätzen*, Master's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2018.
- Salina Huck. *Kontaktlose Messung des Atmungssignals und der Atemrate unter Nutzung von 3D Daten*, Master's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2018.
- Bálint Kovács. *Investigation in 3D image Reconstruction regarding a Small Baseline Multi-Camera Cetup*, Research Projects, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2017 - 2018.
- Adrian Mohra. *Kntwicklung einer modularen 3D Engine zur Darstellung eines virtuellen Operationsaals*, Research Projects, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2017.
- Paul Wagner. *Konzeptionierung und Umsetzung eins Tester für zeitgleiche EKG und Plethysmogrammsignale*, Research Projects, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2017.
- Elisa Dicke. *Untersuchung von Auswirkungen von Bildinterpolationen nach einer Rotation auf Triangulationsfehler*, Master's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2017.
- Christian Marzi. *Konzeptionierung und Aufbau eines experimentellen Aufnahmesetups für operative, mikroskopische Bilddatensätzen zur 3D Rekonstruktion Masterarbeit*, Master's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2017.