Semantic Segmentation of Human Torso Region for Laparoscopic Surgery

By

Salman Maqbool

00000119600

Supervisor

Dr. Hasan Sajid

Department of Robotics and Artificial Intelligence

School of Mechanical and Manufacturing Engineering (SMME)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

February 2018

Semantic Segmentation of Human Torso Region for Laparoscopic Surgery

By

Salman Maqbool

00000119600

Supervisor

Dr. Hasan Sajid

Co-supervisor

Dr. Osman Hasan

A thesis submitted in conformity with the requirements for the degree of *Master of Science* in Robotics and Intelligent Machines Engineering

Department of Robotics and Artificial Intelligence
School of Mechanical and Manufacturing Engineering (SMME)
National University of Sciences and Technology (NUST)
Islamabad, Pakistan

February 2018

FORM TH-4

National University of Sciences & Technology

MASTER'S THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by (Student Name & Regn No.) Salman Maqbool (00000119600) **titled** "Semantic Segmentation of Human Torso Region for Minimal Invasive Surgery" be accepted in partial fulfillment of the requirements for the award of Masters in Robotics and Intelligent Machines Engineering degree with (_____grade).

Examination Committee Members

1. Name Dr. Osman Hasan Signature:_____

2. Name Dr. Yasar Ayaz Signature:_____

3. Name Dr. Muhammad Naveed Signature:_____

Supervisor's name: Dr. Hasan Sajid Signature: _____

Date: _____

_____ _____ _____

Head of Department Date Date

COUNTERSIGNED

Date: _____ _____

Dean/Principal

Declaration

I, *Salman Maqbool* declare that this thesis titled "Semantic Segmentation of Human Torso Region for Laparoscopic Surgery" and the work presented in it are my own and has been generated by me as a result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a Master of Science degree at NUST
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated
3. Where I have consulted the published work of others, this is always clearly attributed
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work
5. I have acknowledged all main sources of help
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself

---

Salman Maqbool,
00000119600

Certificate of Plagiarism

It is certified that MS Thesis titled **Semantic Segmentation of Human Torso Region for Laparoscopic Surgery** by *Salman Maqbool* Registration Number *00000119600* has been examined by us. We undertake the follows:

1. Thesis has significant new work/knowledge as compared already Published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.

2. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results, or words of others have been presented as author own work.

3. There is no fabrication of data or result which have been compiled/analyzed.

4. There is no falsification by manipulating research materials, equipment, or processes, or changing or omitting data or result such that the research is not accurately represented in the research record.

5. The thesis have been checked using TURNITIN (copy of originally report attached) and found within limits as pre HEC Plagiarism Policy and instructions issued from time to time.

Name and Signature of Supervisor,
Dr. Hasan Sajid

Copyright Notice

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the author and lodged in the Library of SMME, NUST. Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.

- The ownership of any intellectual property rights which may be described in this thesis is vested in SMME, NUST, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of SMME, which will prescribe the terms and conditions of any such agreement.

- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of SMME, NUST, Islamabad.

Dedicated to *Abdul Sattar Edhi*.

Abstract

We propose a deep learning based semantic segmentation algorithm to identify and label the tissues and organs in the endoscopic video feed of the human torso region. Our contributions in this project are two-fold: first, we contribute an annotated dataset created from actual endoscopic video feed of surgical procedures, and secondly, we propose a deep neural network for semantic segmentation. To cater to the low quantity of annotated data, we propose unsupervised pre-training and data augmentation. The trained model is evaluated on the independent test set of the proposed dataset. This thesis serves as the first step towards autonomous minimal invasive surgery.

Keywords: *semantic segmentation, deep learning, laparoscopic surgery, convolutional neural networks*

Acknowledgments

I would like to thank Dr. Hasan Sajid for his guidance throughout the project, and for accepting me as his student despite all the problems which were prior to the thesis. Not only has he been the technical supervisor and mentor for me in Computer Vision and Deep Learning; he has been an inspiration on what one can achieve with hard work, consistency, and ambition. Dr. Hasan has been more than a supervisor and I owe a lot of my success to him. Dr. Hasan provided me with a lot of opportunities outside of my thesis too, which I believe contributed a lot to my technical grooming and ultimately getting positions at CERN.

Secondly, I am thankful to Dr. Faisal Shafait and Dr. Ahmad Salman from SEECS who also took me as a student to work on their projects. It was the start of deep learning for me, having switched from mechanical engineering, and their guidance and mentorship led to opening up of a lot of opportunities and avenues for me later on.

Third, I am thankful to my parents for always thinking the best for me and guiding me to always strive to be a better human being, to be helpful to others, and to work on something which creats a positive impact in the world.

I am thankful to Aqsa, my project partner, for without her help, this would have been much more difficult. And for her understanding my busy routine at CERN. I would especially like to highlight that the dataset we prepared was my joint work with her.

Lastly, I would like to thank Usman Ayub Sheikh for the LaTeX template for this thesis.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

HMM	Hidden Markov Model
CNN	Convolutional Neural Network
ML	Machine Learning
DL	Deep Learning
MICCAI	Medical Image Computing and Computer Aided Intervention
BOW	Bag of Words
SVM	Support Vector Machine
FPS	Frames Per Second
FCN	Fully Convolutional Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
IoU	Intersection over Union
FC	Fully Connected
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
CRF	Conditional Random Field

SGD	Stochastic Gradient Descent
DSC	Dice Similarity Coefficient
AI	Artificial Intelligence

Chapter 1

Introduction

Deep Learning has revolutionized a diverse variety of domains these days, from Computer Vision to Natural Language Proecssing to Reinforcement Learning. These in turn have application from self driving cars to healthcare to document analysis to advancement of Artificial General Intelligence. Just like large scale datasets like ImageNet [4] brought about significant advances in image classification, datasets like the PASCAL VOC [5] and MS COCO [6] have made possible training of powerful Deep Learning models for Object Detection and Semantic Segmentation. Especially considering the domain of autonomous vehicles, datasets like KITTI [7] and CamVid [1] combined with the state of art Deep Neural Networks have made large strides possible. Self-driving cars, powered by Deep Learning, would be in production and use within the next 5-10 years. We expect that Aritificial Intelligence (AI) will similarly bring significant advances in healthcare and surgical procedures as well. To this end, we propose this thesis as a first step towards autonomous robotic surgeries.

1.1 Motivation

Deep Learning has already made possible more powerful methods in healthcare, including predicting diabetec retinopathy [8] and cardiovascular risk factors [9] from retinal images, and breast cancer detection [10] from pathological images. However, little work has been done in the domain of automating robotic surgical procedures. While complex systems like the DaVinci surgical robot have made surgical operations more precise and quicker, lead to less blood loss than conventional surgeries, and lead to quicker patient

recovery times; they require a high level of skill and domain knowledge.

Recent advances in AI and Robotics can, however, be used to automate such procedures, which can result in even more precise procedures (removing human error), as well as free up more time for doctors and medical practitioners to give to other pressing issues and medical research. Our proposed work aims to set the foundation for such work.

1.2 Problem Definition

Our work focuses on the broad category of Scene Understanding. In particular, it focuses on pixel level scene understanding; which can be framed as a pixel level identification or classification problem i.e. Semantic Segmentation. We show an example of the problem domain in Figure 1.1.

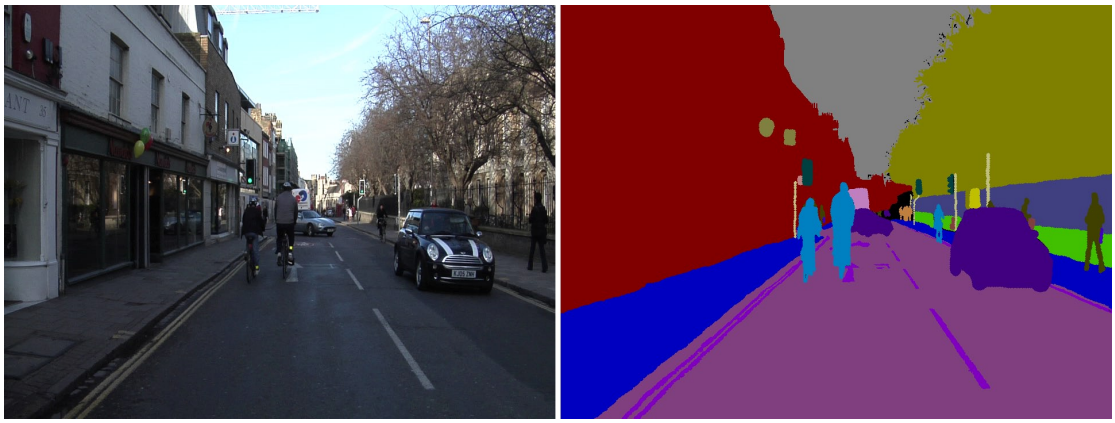

Figure 1.1: The Semantic Segmentation problem, where a mapping needs to be found from an input image (Left) to it's Semantic Segmentation mask (Right) i.e. where all pixels are assigned to their respective categories. The example is from the CamVid dataset [1].

Previous work combining Semantic Segmentation and Robotic surgeries has been limited to instrument segmentation. Deep Learning wise, the work has been limited to Tool Presence Detection (multi-class classification) and Surgical Phase Identification. However, for automating surgical procedures, this is not enough. In particular, at the very least, we need to precisely identify the different organs, instruments, and other entities present in surgical images and videos. Once we have a good understanding of the presented scene, only then can we think of automating the procedure. Our work presents the problem where we have to identify, at a pixel level, every entity present in

images of such procedures. Figure 1.2 shows an example.

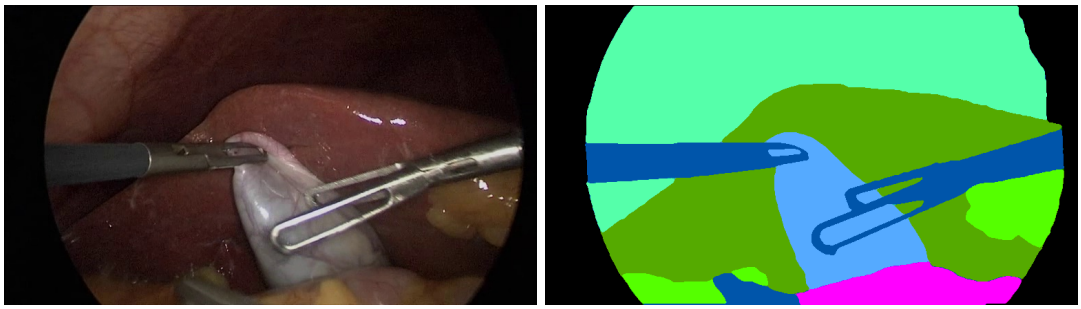

Figure 1.2: Surgical Scene Segmentation example, where the input (Left) has a corresponding pixel level mask where each pixel is assigned a class label (Right).

To this end, our contributions in this thesis are:

1. Proposal of a novel problem in the laparoscopic surgical imaging domain
2. A novel dataset, miccaiSeg, which can be used to train and evaluate any proposed algorithms for the task.
3. An Encoder-Decoder Convolutional Neural Network architecture for addressing the problem.
4. Baseline results which the research community can build upon and improve.

Chapter 2 gives an overview of the existing literature in the domain, and Chapter 3 discusses the proposed dataset and neural network architecture. In Chapter 4 we present the results of our network on the dataset. Lastly, in Chapter 5, we conclude the thesis and outline proposals for future work.

Chapter 2

Literature Review

We focus on two aspects in the Literature Review: Laparoscopic Image and Video Analysis, and Semantic Segmentation.

2.1 Laparoscopic Image and Video Analysis

Prior work done in Laparoscopic image and video analysis focused primarily on three different aspects:

1. Surgical Phase Segmentation
2. Tool Presence Detection
3. Surgical Tool Segmentation

2.1.1 Surgical Phase Identification

Surgical phase identification (also referred to as surgical phase segmentation) refers to identification of the temporal phase of a surgical procedure. A surgical operation is sub-categorized into different phases of the surgery. This has applications in surgical coaching, education, automated and assited surgical procedures, and post surgical analysis of the operation.

Most methods for surgical phase identification have used some variant of Hidden Markov Models (HMMs). Recently, Volkov et al. [11] proposed a method which uses color, organ position, shape (for instruments), and texture features to obtain a Bag-of-Words (BOW)

representation of frames in surgical videos. They use multiple binary Support Vector Machine (SVM) classifiers for each phase to classify frames. They then use a temporal HMM to correct the initial SVM predictions. They used videos of the Laparoscopic Vertical Sleeve Gastrectomy procedure to segment them into seven distinct phases. They obtain 90.4 % accuracy using SVMs, and improve it to 92.8 % with the HMM correction.

The M2CAI 2016 Surgical Workflow challenge held as part of the Medical Image Computing and Computer Aided Intervention (MICCAI) conference in 2016 introduced the TUM LapChole dataset [12]. The dataset contains 20 videos (15 Training and 5 Test) of the Laparoscopic Cholecystectomy procedure. The videos are annotated and categorized into 8 distinct phases, namely:

1. Trocar placement
2. Preparation
3. Calot's triangle dissection
4. Clipping and cutting of cystic duct and artery
5. Gallbladder dissection
6. Gallbladder packaging
7. Cleaning and coagulation of liver bed (haemostasis)
8. Gallbladder retraction

They also provided baseline results for the challenge by using an AlexNet [13] model trained and tested on 1 frame extracted per second. They later used a sliding window approach to correct misclassifications by taking the majority vote among the last 10 predictions. The baseline results were average Jaccard Index, average Precison, and average Recall of 52.4 %, 65.9 %, and 74.7 % respectively.

The challenge winning entry from Jin et al. [14] used a Recurrent Convolutional Network model, EndoRCN. They used a 50 layer ResNet [15] trained for classification into the eight categories as a visual feature extractor. Secondly, they used the current frame and the previous 2 frames to extract the visual features using the ResNet model. The 3 extracted features were fed sequentially to a LSTM model which predicted the phase.

Post-processing using sequential consistency was performed to further improve the predictions. The authors achieved a Jaccard Index score of 78.2.

Another important work in this domain was proposed by Twinanda et al. [16], where they introduced another dataset, Cholec80. The Cholec80 dataset contains 80 videos of the Cholecystectomy procedure, sampled at 1 FPS, where each frame is annotated with the surgical phase information, and addtionally, also the tool presence annotations. The surgical phases are once again divided into 8 distinct categories, while there are tool presence annotations for 7 different surgical instruments (Tool presence annotation are discussed in detail in Section 2.1.2). The authors use a modified AlexNet [13] architecture, which predicts both tool presence in a frame, and uses that, along with the network features, to predict the surgical phase.

2.1.2 Tool Presence Detection

Tool Presence Detection is a multi-class multi-label classification problem, where a mapping is desired from image pixels to a vector representing the presence of surgical tools in the image. This problem can be framed as an image classification problem: a field Machine Learning and Convolutional Neural Networks has dominated in recent years. Tool Presence Detection has applications in automated and assistive surgeries, surgical workflow analysis, and as [16] showed, in aiding phase segmentation.

As discussed earlier in section 2.1.1, the EndoNet architecture [16] was used for joint training of tool presence detection and surgical phase segmentation. This particular work also led to the M2CAI Surgical Tool Detection Challenge held at MICCAI 2016. The challenge dataset, hereby referred to as M2CAI-tool dataset, consists of 15 videos of cholecystectomy procedures, of which there are 10 training videos and 5 test videos. The dataset contains tool presence annotation of the following 7 tools:

- Grasper: *Used to grasp and maneuver the different organs and tissues*
- Bipolar: *Used to seal tissues to stop haemorrhages or blood loss*
- Hook: *Used to burn tissue for ease of later dissection*
- Clipper: *Used to seal tissues and blood vessels before dissection*
- Scissors: *Used for tissue dissection*

- Irrigator: *Used to introduce saline water in case of bleeding/bile. Also used as a fluid suction*
- Specimen Bag: *Used to collect and bring the dissected organ out of the body*

Not surprisingly, the challenge winning entry by Raju et al. [17] used Convolutional Neural Networks for the task. They used an ensemble of the popular VGGNet [18] and GoogLeNet [19] architectures to achieve a mean Average Precision (mAP) of 63.7 % on the 5 videos in the test set.

The M2CAI-tool dataset is worth mentioning, again, because our work builds upon this dataset for semantic segmentation.

2.1.3 Surgical Tool Segmentation

Surgical Tool Segmentation is identifying, at a pixel level in an image, where the surgical tool(s) lie. Surgical Tool Segmentation is one of the important research areas explored since a few years in the domain of computer-assisted surgical systems. This is important since it can provide feedback and other guidance signals to the surgeon. This also helps immensely in surgeries requiring higher precision. Segmenting the tool is important at a pixel level becasue of the critical nature of surgical procedures. Accurate tool segmentation can then lead to accurate tool localization and tracking. This step is also essential for automating surgeries, but not enough, as we also need information about non-instrument part of the scene. Nevertheless, since our work is an extension of this, we would first like to discuss some approaches for the task.

Traditionally, image processing techniques were the dominant approach for the task. Since the scenario has changed with the success of Deep Learning algorithms, we would focus more on those. But we would describe one of the methods for reference. Doignon et al. [20] introduced a method based on a combination of various image processing techniques, including the use of hue, saturation, edge detection, region growing, and using shape features to classify regions in an image.

More recently though, García-Peraza-Herrera et al. [21] proposed a real-time tool segmentation method which uses Fully Convolutional Networks (FCNs) along with Optical Flow based tracking to segment surgical instruments in videos. We will go through FCNs in more detail in Section 2.2. Due to hardware limitations of running FCN inference

in real-time, they use OpticalFlow tracking and assuming somewhat rigidity of the tool and scene for a few frames, they compute affine transformation of the new segmentation mask with respect to the previous one. The segmentation mask are updated as the FCN computes the results; enabling real-time segmentation. However, with today's hardware and efficient Deep Learning architectures, using a purely Deep Learning system for real-time segmentation is very much possible. More recently, García-Peraza-Herrera et al. [22] introduced ToolNet, a modified version of the FCN. They introduced 2 different architectures, one which aggregated predictions across multiple scales before calculating the loss, and another which incorporated a multi-scale loss function. They also used the Dice Loss [23, 24], which has shown to be effective for semantic segmentation, especially across unbalanced classes. We also incorporate the Dice Loss in our work, which is explained in Chapter 3. They also used Parametric Rectified Linear Units (PReLU) [25], an adpative version of the Rectified Linear Unit (ReLU) as the activation function. They achieve a Balanced Accuracy of 81.0 % and a mean Intersection over Union (IoU) of 74.4 % on the Da Vinci Robotic (DVR) dataset, which was part of the MICCAI 2015 Endoscopic Vision (EndoVis) Challenge.

Attia et al. [26] proposed a Hybrid CNN-RNN Encoder-Decoder network for surgical tool segmentation. They used a 7 convolution layered CNN to extract feature maps from the input image. Using just an Encoder-Decoder network produced coarse segmentation masks. To cater to this and to account for spatial dependencies among neighboring pixels and to enforce global spatial consistency, the authors used 4 Long Short Term Memory (LSTM) Recurrent Neural Network (RNN) layers in sequence on the produced encoder feature maps. They then used a decoder network to upsample the feature maps into the final segmentation masks. They achieved a balanced accuracy of 93.3 % and an IoU of 82.7 % with their method on the MICCAI 2016 Endoscopic Vision (EndoVis) Challenge Robotic Instruments dataset.

Another important advance in surgical tool segmentation was the segmentation of the tool into it's constituent parts i.e. the tool shaft and the tool's manipulator. This tranforms the binary classification/segmentation problem into a 3-class classification problem, where the 3rd category is Background. Pakhomov et al. [27] proposed a 101-layered ResNet [15] model which is casted as a FCN for semantic segmentation. They use Dilated Convolutions [28] to reduce the down-sampling induced by Convolutional layers without padding. Dilated (also called Atrous) convolutions have proven useful for

semantic segmentation as it allows larger receptive fields while keeping the number of network parameters low. They achieved the state of the art results (at the time of their paper) on the MICCAI 2015 EndoVis Challenge Robotic Instruments dataset at 92.3 % Balanced Accuracy for Binary Classification/Segmentation. They also reported results for the multi-class segmentation case which can be accessed in their publication [27].

Since instrument segmentation and localization are interdependant tasks, Laina et al. [29] utilize that for concurrent segmentation and localization of surgical instruments. Additionally, they frame localization as a heatmap regression problem, where they use landmark points on the instruments with a Gaussian centered around them to generate the groundtruth. They then regress for the heatmaps (one per landmark), which represent the confidence of each pixel to be in the proximity of the groundtruth landmark. This approach makes training easier and stable over regressing over 2D (x, y) coordinates of the landmark points. The authors train jointly for both segmentation and localization, which helps in improving performance for both tasks. They also use a multi-class segmentation approach similar to [27], but with 5 different classes namely: Left shaft, Right shaft, Left Grasper, Right Grasper, Background. They obtain a Balanced Accuracy of 92.6 % on the MICCAI 2015 EndoVis Challenge.

As we can see from the above approaches, Deep Neural Networks, and CNNs in particular, have been the de facto approach for various tasks in Laprascopic Image and Video analysis in the previous few years. And rightly so since their success in this domain builds upon the success of Deep Learning in general over the last few years. It should also be noted that joint training for multiple objectives has shown to be helpful in training for all the objectives.

That said, all the above methods focus on instrument segmentation. For a better and more complete understanding of a robotic scene, especially considering autonomous robotic surgeries, this is insufficient. We need to know the precise location of not just the tools, but also the organs. Till date, to the best of our knowledge, no such approach for dense instrument and organ segmentation has been explored in literature.

2.2 Semantic Segmentation

The second part of our Literature Review focuses on Semantic Segmentation, which is a pixel level classification problem; and has been a landmark domain in Computer Vision research. But before delving into the Semantic Segmentation literature, it is worth exploring Convolutional Neural Networks in general, which make for the backbone of today's Semantic Segmentation models. We will also go in more detail in Deep Learning terminology in this section.

2.2.1 Convolutional Neural Networks

The field of Deep Learning has a long and interesting history. We refer the reader to [30] for an overview of Deep Learning over the recent years, and to [31] for a comprehensive history and overview of the domain.

The first attempt at a Convolutional Neural Network was by Fukushima in 1982 in his famous Neocognitron paper [32]. The architecture was inspired by Hubel and Weisel's Nobel Prize Winning experiments with the cat visual cortex, where they discoverd that specific neurons in the cat visual cortex responded to specific patterns showed to the cat. Later, LeCun et al. [33, 34] applied the backpropagation algorithm to make CNNs learn by gradient-based optimization to recognize handwritten digits. This was an important breakthrough as it demonstrated that CNNs can be optimized much more easily than before.

LeNet is a 7-layered Neural Network comprising of Convolutional Layers, Sub-sampling layers (now called Polling layers), and Fully Connected Layers. 2D Convolution, in general, is an operation which takes the dot product of a filter of size k with the input (in Computer Vision the input is mostly images) in a sliding window manner (where the amount of sliding is determined by a stride) to produce an output feature map. Figure 2.1 demonstrates a convolution operation with a 2D matrix (which is analogous to a grayscale image).

Convolutional Neural Networks generally comprise of a large number of such convolutional filters, whose parameters (or weights) are learnt gradient based optimization algorithms such as gradient descent. The gradient, in this case, is defined as the gradient of the parameters with respect to the loss; where the loss is a pre-defined cost

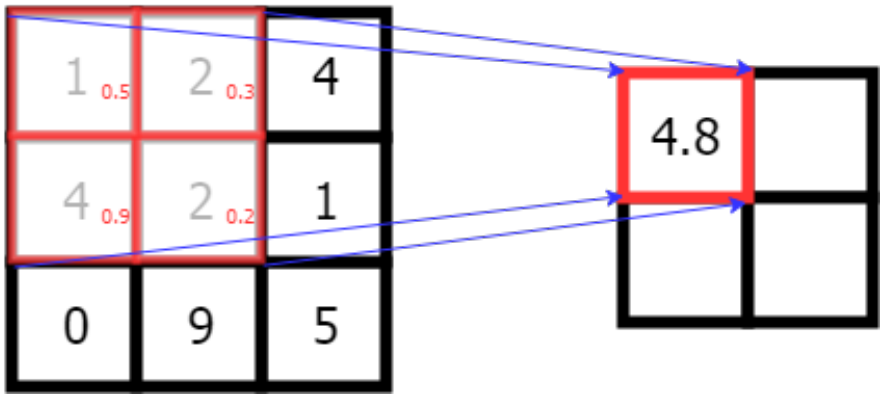

Figure 2.1: Example of a convolution operation.

function which measures the disagreement between the network predictions and the ground truth.

Another important component of the LeNet were the subsampling layers. These layers subsample the input into a lower dimensional output, which reduces the network parameters while retaining the most important information. Secondly, pooling allows for translation invariance as any information from any neurons which respond to some specific feature in the input is retained irrespective of where the feature is found in the input. This has some drawbacks, and is considered indesirable nowadays, but nevertheless, subsampling and pooling layers remained an important part of CNN architectures till a few years ago. Figure 2.2 depicts the max pooling operation, where the maximum element is taken from each sampling window.

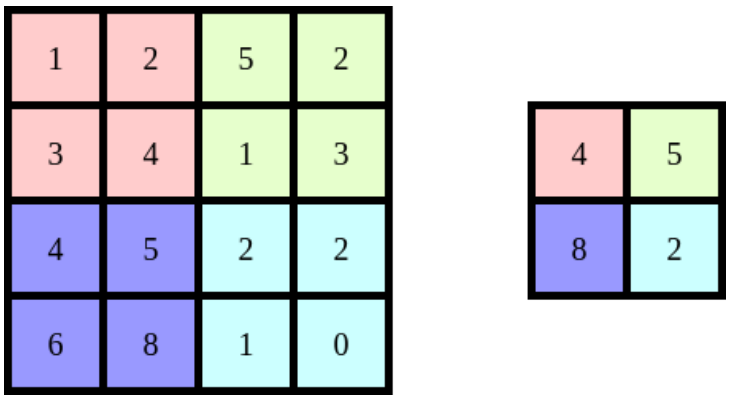

Figure 2.2: The Max Pooling operation with Kernel size 2 and Stride 2 . The input is shown on the left while the output is on the right (Image inspired from [2]).

Fully Connected (FC) layers (also called Linear or Dense layers) are the typical Neural Network layers where each unit in the input has a connection with every unit in the

output. The standard neural networks are made up of a number of Fully Connected layers. A fully connected layer is depicted in Figure 2.3.

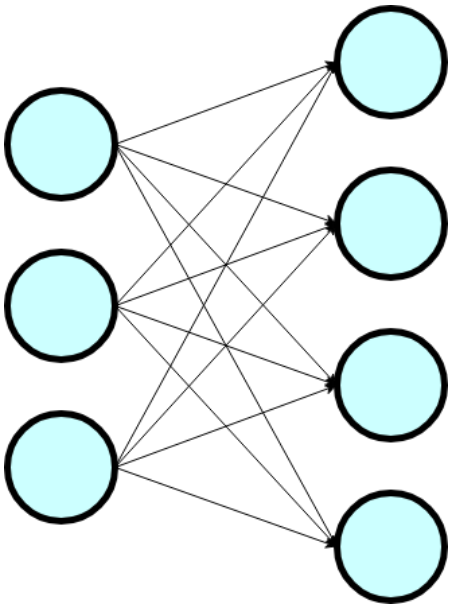

Figure 2.3: A Fully Connected Layer, shown connecting 3 neurons in the input to 4 neurons in the output. The arrows represent the learnable network weights.

More recently though, one of the main triggers for the current Deep Learning era was Alex Krizkevsky et al.'s ImageNet [4, 35] winning work [13], where they used Convolutional Neural Networks to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [35] by a large margin over traditional Machine Learning methods. They introduced AlexNet, a 7-layered CNN, which they leveraged GPUs for faster training times over the massive dataset. They used ReLUs, Dropout [36], and parallelization over GPUs to achieve a then State of the Art top-5 error rate of 15.3 % on the ILSVRC.

Soon after, various more powerful CNN architectures emerged [15, 18, 19, 37], leading to quicker, simpler, more efficient, and more accurate training of Convolutional Neural Networks; and leading to significant advancement in almost every domain, especially Computer Vision.

2.2.2 Semantic Segmentation Networks

Semantic segmentation is the pixel level labeling/classification for any image/video. It is the natural step after success of several deep learning based object detection networks [38–43], where objects are located by a bounding box. Object detection at pixel level,

or getting an accurate object mask, is critical for many applications such as self-driving cars, and especially in our case of Robotic Surgeries.

Long et al. [44] introduced the first popular end-to-end trainable Deep Learning architecture for Semantic Segmentation, the Fully Convolutional Network (FCN).

FCN is a deep CNN which uses a series of Convolution and Pooling layers to generate feature maps. The feature maps are then unsampled using Fractionally-Strided Convolutions. Fractionally-Strided Convolutions or Transpose Convolutions (also sometimes wrongly referred to as Deconvolution) zero-pad the input feature map between the pixels, where the number of zeros is the scale factor (k) - 1. A regular convolution is then performed on this fractionally padded input. This results in a learnable and differentiable upsampling filter. An example of this is given in figure 2.4. In addition to Fractionally Strided Convolutions, the authors combine information across different layers (which results in information sharing across different scales). Lastly, they use an n-way softmax for each pixel for prediction, where n is the number of classes.

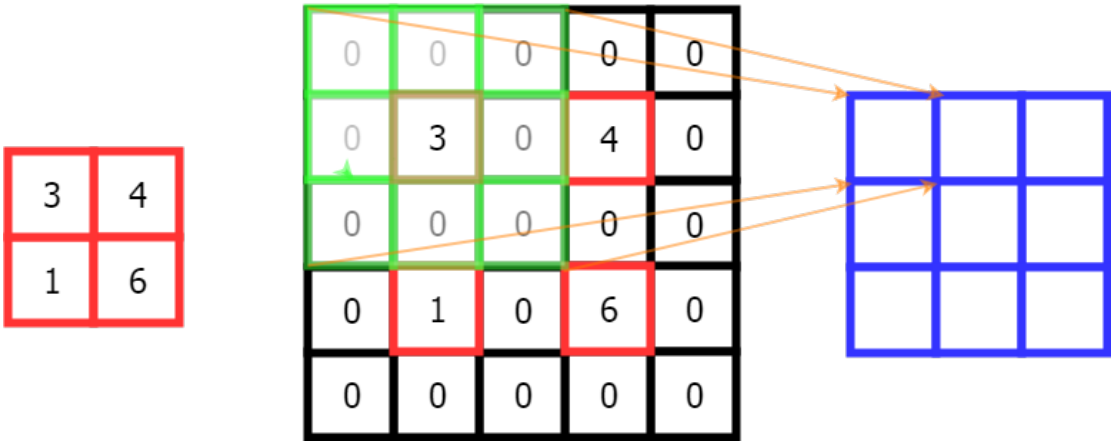

Figure 2.4: An example of the Fractionally Strided Convolution operation. The input feature map is shown on the left in Red. The middle image shows the zero-padded input image, with the convolution kernel shown in Green. The upsampled output is shown on the right in Blue.

Soon after the release of FCN, Badrinarayanan et al. [45] proposed an Encoder-Decoder network for Semantic Segmentation called SegNet. The Encoder part of the network is identical to the VGG network [18], where the max pooling indices for each layer are stored for upsampling later. In the Decoder part of the network, feature maps are successively upsampled using the corresponding max pooling indices into sparse feature

maps. These sparse feature maps are then convolved with learnable filters to obtain dense feature maps, and ultimately a semantic segmentation of the input.

Another Semantic Segmentation architecture worth discussing is the U-Net architecture [3], which was proposed for biomedical image segmentation. The main feature of the proposed U-Net architecture is feature concatenation (or information sharing) from earlier in the network to the later layers. This helps retain low level features like edges, which helps in obtaining sharper segmentation masks. The architecture has been successful and popular for biomedical image segmentation tasks.

Most popular (and powerful) Semantic Segmentation models use Conditonal Random Fields (CRFs) [46] to post-process and refine the network predictions. This makes the network step-wise and not end to end trainable. To circumvent this, Zheng et al. [47] introduced the CRF-RNN model, where they modeled the CRF component of the network as a Recurrent Neural Network. This in turn resulted in an end-to-end trainable network capable of semantic segmentation with a trainable CRF component.

More recently, another popular architecture, the Fully Convolutional DenseNet [48] was proposed by Jégou et al. This architecture tries to remove the CRF out of the equation by modifying the DenseNet architecure [49] for Semantic Segmentation. [49] proposed the DenseNet architecture which is composed of Dense blocks (which can be said as an extension to the Residual Blocks in [15]). The dense blocks function by concatenating the outputs of all previous layers (including the first input) with the output of the current layer. The problem with such an architecture, especially for semantic segmentation as it requires a Decoder network as well, is the explosion in the number of feature maps as we go deeper in the network. To cater to this problem, [48] do not concatenate the Dense Block input to the output in the Decoder Network. They also use skip connections to combine information from the Encoder network with the Decoder network.

Chapter 3

Proposed Work

Our work builds upon the methods for Laparoscopic Image and Video Analysis and the general success of Convoluion Neural Networks for Semantic Segmentation, as discussed in Chapter 2. Since no dataset for the task existed at the initiation of the project, we made and annotated a dataset for semantic segmentation of robotic surgical scenes. Section 3.1 describes our work on the miccaiSeg dataset, while section 3.2 details our proposed method.

3.1 The miccaiSeg dataset

The proposed miccaiSeg dataset is my joint work with Aqsa Riaz. Our proposed miccaiSeg dataset is an extension of a small subset of MICCAI 2016 Surgical Tool Detection dataset [16] (M2CAI-tool). The M2CAI-tool dataset, described briefly earlier in Section 2.1.2, consists of a total of 15 videos, which are divided into 10 training videos and 5 test videos. Each video has a tool presence annotation every 25 frames i.e. at 1 FPS. There are a total of 7 tools as detailed in Section 2.1.2. However, before moving on to the miccaiSeg dataset, let's consider our annotation methodology.

3.1.1 Annotation Methodology

Firstly, we had to do a study of different laparoscopic surgical procedures. The two we considered the most were Appendectomy (Appendix removal) and Cholecystectomy (Gall Bladder removal). After study of the procedures and available videos, we eventually settled for Cholecystectomy because of the availability of the open-source M2CAI-

tool dataset. We studied the region's anatomy, the procedure details, and the different organs and surgical instruments involved in the procedure.

Thereafter, we started to investigate the different annotation procedures and conventions to label data for Semantic Segmentation. The most popular conventions are using Json [6] or XML [5] files to store coordinates of the polygons forming an object. However, since we needed more accurate annotation which often involved circular regions, we didn't go for that. We instead relied on a relatively naive approach of generating colored masks for the different object categories.

For tools and software for generating semantic segmenation annotations, LabelMe [50] is a popular online tool for generating polygonal annotations of object masks. However, due to reasons mentioned earlier, and for further ease of annotation, we used a semi-supervised annotation tool based on MegaPixels [51]. We developed a tool in MatLab based on MegaPixels, which segments the region into distinct parts. However, sometimes the tool produced regions which were not representative of segmentation boundaries. For such images, we manually annotated the difficult regions or region boundaries using Microsoft Paint or Photoshop first before passing it to the annotation tool. Overall, using this reduced the annotation time.

3.1.2 Dataset Details

We subsampled 307 images from Videos 1 and 2 of the M2CAI-tool training set and annotated them at a pixel level into various different categories and sub-categories as shown below (The values in square brackets show the [R, G, B] value assigned to pixels of this category):

- **Organs:**
    1. Liver [85, 170, 0]
    2. Gallbladder [85, 170, 255]
    3. Fat [85, 255, 0]
    4. Upperwall [85, 255, 170]
    5. Intestine [255, 0, 255]
- **Instruments:**

1. Grasper [0, 85, 170]
2. Bipolar [0, 85, 255]
3. Hook [0, 170, 85]
4. Scissors [0, 255, 85]
5. Clipper [0, 255, 170]
6. Irrigator [85, 0, 170]
7. Specimen Bag [85, 0, 255]
8. Trocars [170, 85, 85]: *Provide an opening to insert the surgical instruments.*
9. Clip [170, 170, 170]: *The clips applied by the Clipper to seal the blood vessels.*

- **Fluids:**
  1. Bile [255, 255, 0]
  2. Blood [255, 0, 0]
- **Miscellaneous:**
  1. Unknown [170, 0, 85]: *Used as a label for pixels which are indiscernable for the annotator.*
  2. Black [0, 0, 0]: *Used as a label for the surrounding region in the image which is not visible due to the trocar limiting the camera field of view.*
- **Artery** [170, 0, 255]

In total, we annotated a total of 5 different organs, 9 different instruments (of which Trocars and Clip are different from the Tool presence annotation in M2CAI-tool dataset), 2 fluids, 2 miscellaneous categories, and the artery. The annotations were made considering further use of this information for autonomous robotic surgeries. For example, the Clip needs to be identified in the videos to be able to learn where and when to place it. Similarly, the artery needs to be identified since it needs to be sealed. Blood loss indicates potential use of the bipolar to seal any open incisions, while Bile indicates that the irrigator potentially needs to be used for sterilization and cleanup. The other categories are pretty much self-explanatory.

During the image saving procedure, due to the wrong choice of image format (JPEG), we ran into issues further down the stream where we discovered that pixels near object

boundaries are smudged (and don't have the right RGB value). To avoid the costly re-annotation procedure, we performed (for each pixel) a nearest neighbor search in the RGB space (from all the 19 categories) to adjust the pixel value to the nearest RGB value from all the 19 categories. Afterwards, we applied a 5 x 5 median filter to remove the resultant salt and pepper noise from the images. This obviously effected the annotation quality a bit, but we proceeded to use that since the noisy pixels would act as a regularizer during training.

Figure 3.1 shows some sample annotations from our dataset.

3.2 Proposed Network Architecture

Due to the small quantity of annotated data, we focused our efforts on methods which were data efficient or which worked well with smaller datasets like ours. To this end, we propose a minimalist Encoder-Decoder Convolutional Neural Network as shown in Figure 3.2. We will refer to this as the segmentation network.

3.2.1 Network components

Our network has some distinct components, which we identify as:

- The input image
- The convolution layers
- Batch Normalization
- The ReLU non-linearity
- The Convolution Transpose Layers
- Dropout
- Softmax

We briefly go over them in the following sections.

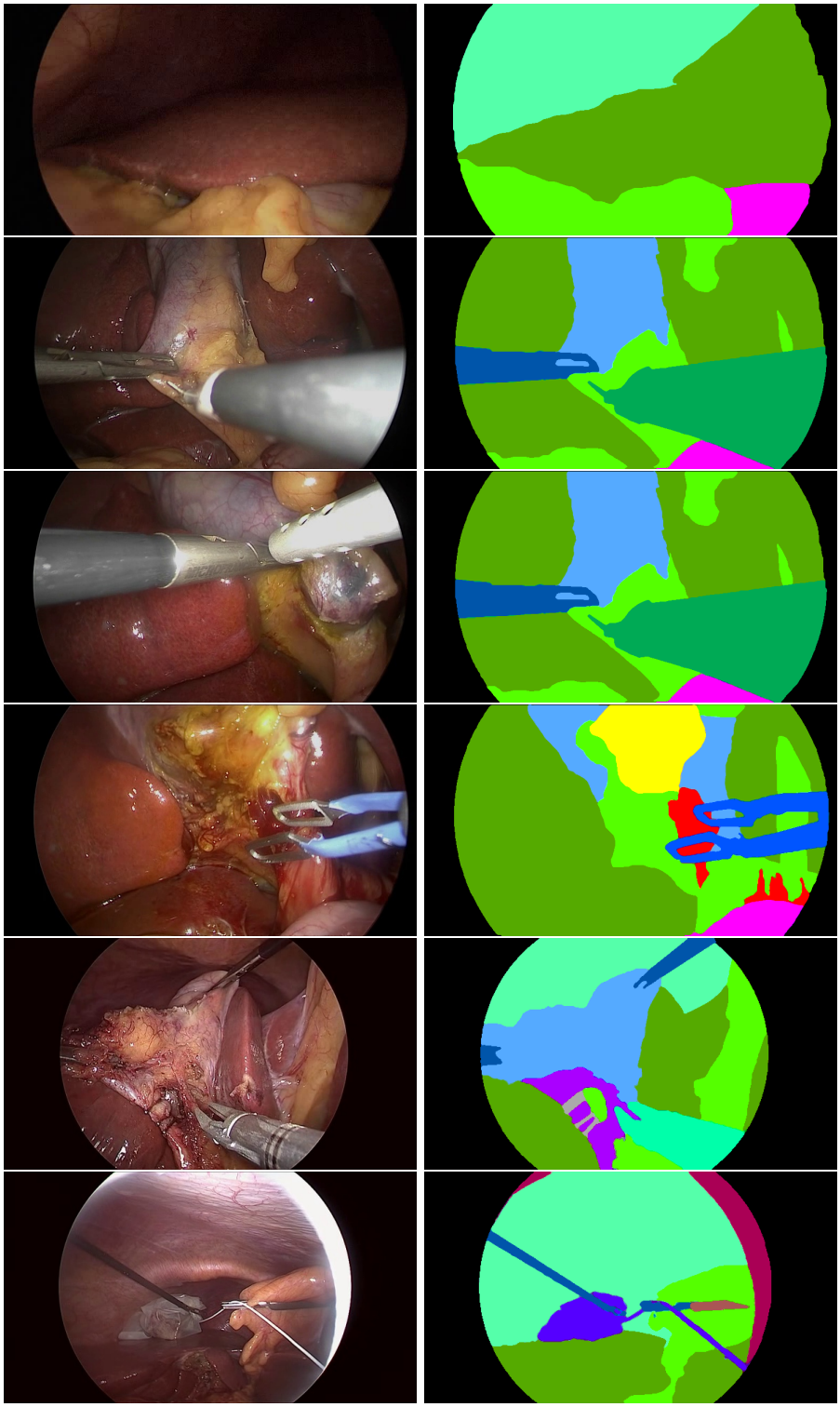

Figure 3.1: Some samples from the miccaiSeg dataset. The left column shows the original images while the right column shows their corresponding groundtruth annotations.

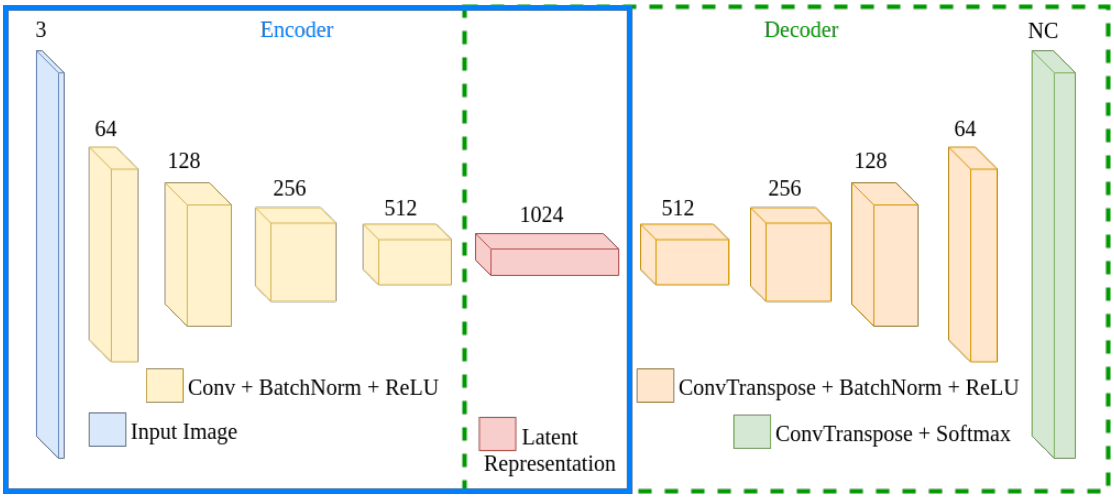

Figure 3.2: Our proposed 10 layer CNN Encoder-Decoder Network. NC means the number of classes, which is equal to the channel dimensions of the network output.

3.2.1.1 Input Image

The input image is resized to 256 x 256 at training time. Due to the small size of our dataset, we perform online 10-crop data augmentation at training time, where we take all 224 x 224 crops from all the 4 corners and the centre, and their horizontal flips (mirror images). We normalize each image by its RGB per-channel mean [0.295, 0.204, 0.197] and standard deviation [0.221, 0.188, 0.182]. These values are computed over all the 581923 frames of the M2CAI-tool training set. Figure 3.3 demonstrated the 10-crop data augmentation.

At test time, the input image is normalized, but we don't use 10-crop data augmentation. In this case, the input image is resized to 256 x 256, but since we don't take any crops, the resolution stays at 256 x 256 and does not go to 224 x 224.

3.2.1.2 Convolution Layers

Convolution layers are composed of a number of learnable convolution filters which operate over the input. Over images, we use 2D convolutions which operate over the spatial dimensions of the image. A convolution is a dot product over a section of the image, and is mathematically defined as:

$$(I * K)_{\text{xy}} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{\text{ij}} \cdot I_{\text{x + i -1, y + j -1}} \tag{3.2.1}$$

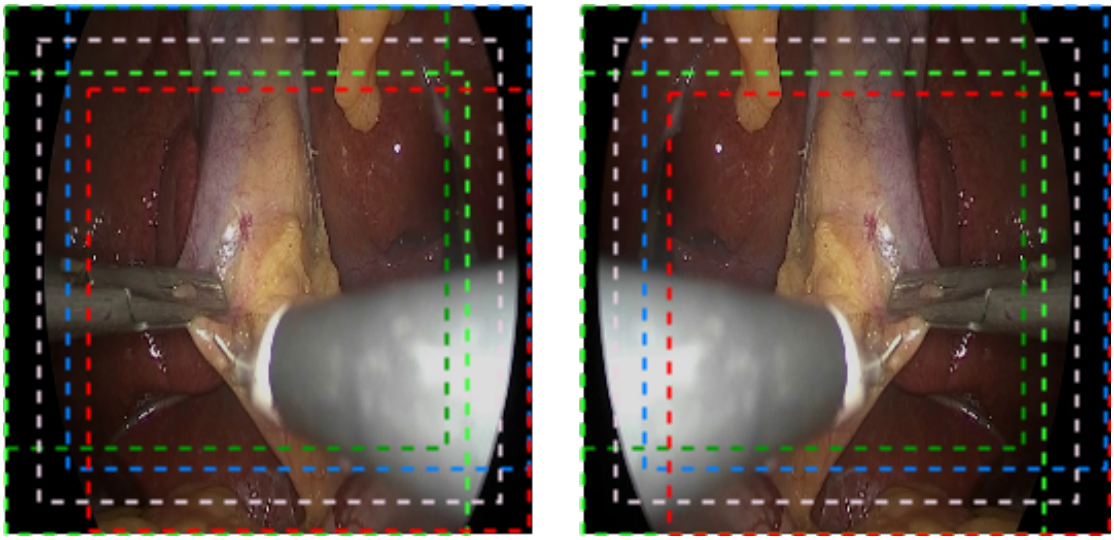

Figure 3.3: 10 crop data augmentation. This is done to increase the number of training samples. The dotted lines show the crop boundaries. The dark green and red boxes have been slightly shrunk in the image for better visibility of the boxes. In practice, all crops are the same size. The original image is shown on the left while it's horizontally flipped version is shown on the right.

where I is the image, K is the convoluion kernel, h is the input height, and w is the input width. For our proposed network, we used convolution layers with 64, 128, 256, 512, and 1024 filters respectively for the Encoder part to obtain the latent representation. We used a kernel size of 4 x 4, stride of 2, and padding of 1 for all the convolution layers in our network except the last encoder layer where we used the same kernel size but with a stride of 1 and no padding.

3.2.1.3 Batch Normalization

Batch Normalization (BatchNorm) was introduced by Sergey and Szegedy [52], which makes weights in later layers of the network more robust to changes in the weights in the earlier layer. Normalizing activations leads to better gradient values and learning. Instead of running values between 0 and 1, though, the output of convolution layers are instead normalized using a running mean and variance. Those values are controlled using learnable parameters γ and β. The output of a batch normalization layer is defined as:

$$y = \frac{x - mean[x]}{\sqrt{Var(x) + \epsilon}} * \gamma + \beta \tag{3.2.2}$$

Batch normalization reduces the covariate shift i.e. the changes in activations of a particular layer. It also induces a slight regularization effect, since we use mini-batches, which adds a slight noise to the activations. We used batch normalization in all layers except the first Encoder layer in our network.

3.2.1.4 ReLU Non-linearity

Using a combination of only convolution layers without any non-linearity would be extremely detrimental to the performance of Neural Networks, as it can only enable learning of linear functions of the input. However, the strength of Neural Networks in practice lies in their ability to learn complex non-linear functions. For that reason, we need to induce non-linearity in the network. Rectified Linear Units (ReLUs) have gained popularity as a non-linearity in recent years in their use in Convolutional Neural Networks since their use in AlexNet [13]. The ReLU non-linearity leads to faster training times over the conventionally used sigmoid non-linearity, as well as helps stabilize the gradients, leading to more stable training. It is graphically shown in figure 3.4 and is mathematically given by:

$$f(x) = max(0, x) \tag{3.2.3}$$

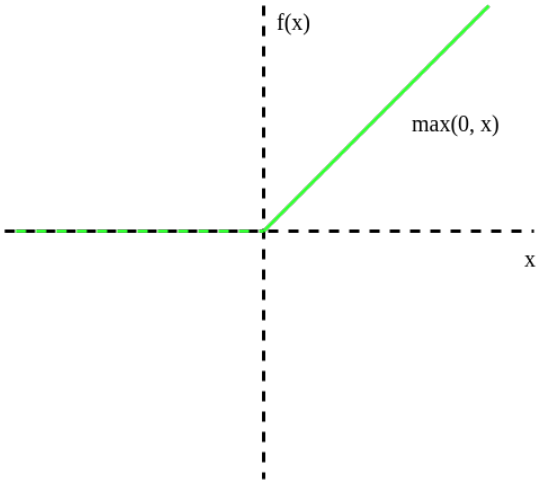

Figure 3.4: The ReLU non-linearity.

3.2.1.5 Convolution Transpose Layers

Once we have obtained the latent representation of our input image after it passes through the Encoder, it then goes through the Decoder network where it is successively upsampled using Convolution Transpose Layers. Convolution Transpose Layers use Fractionally Strided Convolutions described earlier in Section 2.2.2. We mirror the network about the Latent representation, and thus our Convolution Transpose layers use the same hyperparameters as the convolution layers in the Encoder network i.e. the first Decoder layer uses a Stride of 1 and no padding, while all the other layers use Stride of 2 and padding of 1. The Kernel size is 4 x 4 for all the layers. The number of filters used in Convolution Transpose layers are 512, 256, 128, and 64 successively, while the last layer uses filters equal to the number of classes. We also use BatchNorm and the ReLU non-linearity in all Convolution Transpose layers except the last one, where we use a softmax for each pixel. The network output is the same size as the resized input image i.e. 224 x 224 and 256 x 256 for training and inference respectively.

3.2.1.6 Dropout

Dropout [36] is one of the most powerful techniques ever introduced for training deep neural networks. It is standard in almost all deep learning architectures proposed these days. During training, Dropout makes zero, with a certain predefined Dropout probability p, part of the output activations of any layer. This has two powerful effects:

1. It inhibits coadaption of neurons i.e. since during each forward pass, a certain number of neurons are deactivated, it forces other parts of the network to learn discriminating features from the input.

2. It acts as a powerful regularization method since every forward pass, a part of the network is deactivated.

The resultant activations are then scaled up by inverse of the Dropout probability. This is important since at test time, the total activations of each layer would otherwise be significantly larger numerically, which would lead to erroneous predictions. At test time, no neuron is deactivated. This leads to another powerful effect, since now, the network acts as an ensemble of multiple models for which the predictions are averaged. Overall,

Dropout can be termed as one of the most important advances, if not the most important advancement, in Deep Learning in the last few years.

We used Dropout in the first three Decoder layers of our network. We also used the 2D variant of Dropout, which is suitable for Convolutional Neural Networks, and where convolution kernels in the layer are dropped by the pre-specified Dropout probability. In our experiments, we set p as 0.5.

3.2.1.7 Softmax

A softmax converts a series of prediction scores to probabilites. In our case, we use a softmax at the output of our network to calculate the probability of each pixel belonging to each class. The probabilities obtained for each pixel sum to 1. For each pixel, the probability the pixel belongs to class c is given by:

$$P(y_{\mathrm{c}}|x) = \frac{e^{x_{\mathrm{c}}}}{\sum e^{x}} \tag{3.2.4}$$

where x are the raw scores for that pixel for all classes (in our case, it is the output of the last Convolution Transpose Layer), x_c is the raw score for the class c, and y_c is the probability of the pixel belonging to the certain class. The softmax computes the probability of each pixel belonging to each class.

3.3 Training Details

We split our dataset into a training set and a test set, containing 245 images and 62 images respectively. Since our dataset size is rather limited, we explored unsupervised pre-training to learn dataset specific features. We believe this can be particularly helpful for Semantic Segmentation.

We train over the entire M2CAI-tool training set (581935 frames) for image reconstruction over 1 epoch. The network used is the same as the segmentation network, except the last layer of the network uses a Sigmoid instead of a Softmax. We used a learning rate of 0.01, and a batch size of 64 for the training. For the loss function, we used the per-pixel Mean Squared Error loss which is given as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2 \tag{3.3.1}$$

where n is the number of pixels, y is the groundtruth pixel value, and $\hat{y}$ is the predicted pixel value. We initilize our segmentation network with weights from the reconstruction network. We then finetune the network for semantic segmentation. We use the Adam Optimizer [53] to train both our reconstruction and segmentation networks with β_1 and β_2 as 0.9 and 0.999 respectively. Adam stands for Adaptive Moment Estimation and uses a moving average of the gradients to make the update of the network parameters, as opposed to using the gradient of just the current iteration as in vanilla Stochastic Gradient Descent (SGD).

We also used Weight Decay as a regularizer with λ equals 0.0005. Weight decay penalizes large weights and thus allows learning of diverse parameters of more or less equal importance for the network. We used step learning rate decay with the initial learning rate 0.0001, which is halved every 10 epochs. We trained for a total of 90 epochs with a batch size of 2 and used a multi-class pixel-wise Dice loss function.

The Dice Similarity Coefficient (DSC) measures the similarity between two image regions, and the discrete version is given as:

$$DSC = \frac{2 * |Y \cap \hat{Y}|}{|Y| + |\hat{Y}|} \tag{3.3.2}$$

where Y is the ground truth segmentation map, $\hat{Y}$ is the predicted segmentation map. $|\ |$ indicates the number of pixels belonging to a particular class in the groundtruth and the predicted segmentation maps. However, the above function is not differentiable and thus can't be used as a loss function. However, [23] proposed a continuous and differentiable version of the function which can be used directly as a loss function in training Deep Neural Networks. We use that formulation as the loss function of our network. Using the Dice Loss function benefits training in our case as compared to the Cross-Entropy loss.

Chapter 4

Results and Evaluation

We evaluate our proposed network on the test set of the miccaiSeg dataset proposed in Chapter 3. We divide our training and evaluation into two different categories:

1. Single Instrument Class: *We categorize all instruments into one single class i.e. Instruments*
2. All Categories: *We use all categories as outlined in Section 3.1.2*

For evaluation, we report 4 different performance measures, namely Intersection over Union (IoU) (also called Jaccard Index), Precision, Recall, and the F1 score for each class, as well as their mean over all classes.

$$IoU = \frac{TP}{TP + FP + FN} \tag{4.0.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.0.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.0.3}$$

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.0.4}$$

We use a pixel-wise criteria to evaluate the True Positives (TP), False Positives (FP), and the False Negatives(FN). The IoU represents the degree of overlap between the

segmentation regions, benefiting from the True Positives while penalizing both the False Positives and False Negatives. Precision and Recall represent the resilience to False Positives and False Negatives respectively. Finally, the F1 score is the harmonic mean between Precision and Recall and gives a more balanced estimate taking into account both the False Positives and False Negatives.

4.1 Single Instrument Class

These experiments use a subset of all 19 classes, clustering all the instruments into 1 super-category, and merging the Fluid super-category (Blood and Bile) with the Gallbladder class; resulting in a total of 9 classes, namely:

1. Unknown
2. Instruments
3. Liver
4. Gallbladder
5. Fat
6. Upper Wall
7. Intestine
8. Artery
9. Black

Table 4.1 shows the results of our proposed network on the above categories.

As we can see from Table 4.1, our network performs well on the majority classes in the dataset, while its performace suffers on the less dominant classes; especially Intestine and Artery. This is expected since the number of instances of these two classes are quite low. This is also the case for the Unknown class as our network fails to correctly predict that class throughtout the evaluation. This can potentially be explained by the fact that while human annotators didn't get how a particular part of the image should

Class	IoU	Precision	Recall	F1 Score
Unknown	0.00	0.00	0.00	0.00
Instruments	0.73	0.79	0.91	0.85
Liver	0.77	0.84	0.90	0.87
Gallbladder	0.50	0.80	0.58	0.67
Fat	0.53	0.61	0.79	0.69
Upper Wall	0.41	0.65	0.53	0.58
Intestine	0.17	0.80	0.18	0.30
Artery	0.09	0.49	0.09	0.16
Black	0.94	0.96	0.97	0.97
Mean	0.46	0.66	0.55	0.57

Table 4.1: Results of our network on the miccaiSeg with a single instrument class.

be labeled, and hence annotated it as Unknown; the algorithm learns features through which it is able to make a prediction other than Unknown for those image regions.

That being said, the more dominant classes especially Instruments, Liver, and Black perform well. However, the algorithm fails to impress for Gallbladder and Fat classes, which are also common. The algorithm mostly confused the Gallbladder and Instruments class. Figure 4.1 shows some of the predictions for our network, while Figure 4.2 shows some failure cases for our network.

As can be seen from Figure 4.2, most failures are for images which are difficult to discern. Additionally, our network sometimes confuses the Gallbladder for an Instrument due to potentially similar colors and shape.

4.1.1 Comparison with U-Net

U-Net [3], as disussed earlier in Chapter 2, is a popular architecture for Biomedical Image Segmentation. We compare the performance of our network with U-Net. The U-Net architecture was trained from scratch, with the same hyperparameters as our proposed network. We detail the results in Table 4.2.

Our proposed method outperforms UNet on all categories in all evaluation criteria. It also shows that the unsupervised pre-training is especially beneficial for small datasets

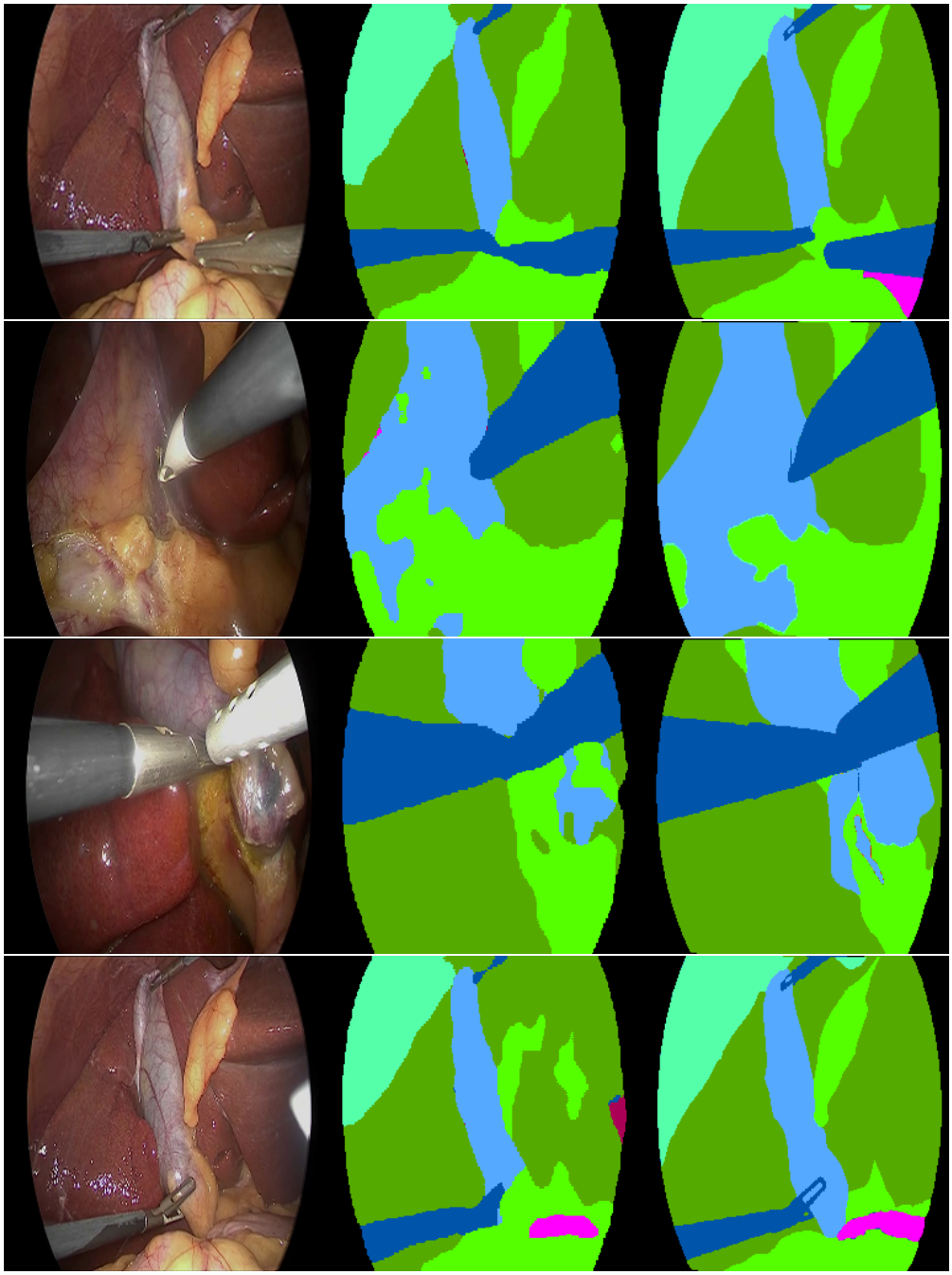

Figure 4.1: Predictions of our network on the miccaiSeg dataset with the single instrument class. The left column shows the original images, the middle column shows the prediction, while the right column shows the corresponding groundtruth.

in Semantic Segmentation. Figure 4.3 shows the result for our proposed method and UNet for a particular image from our test set.

As we can see from Figure 4.3, UNet preserves the lower level features nicely such as edges and shape information, but fails to makes fine predictions globally. We propose

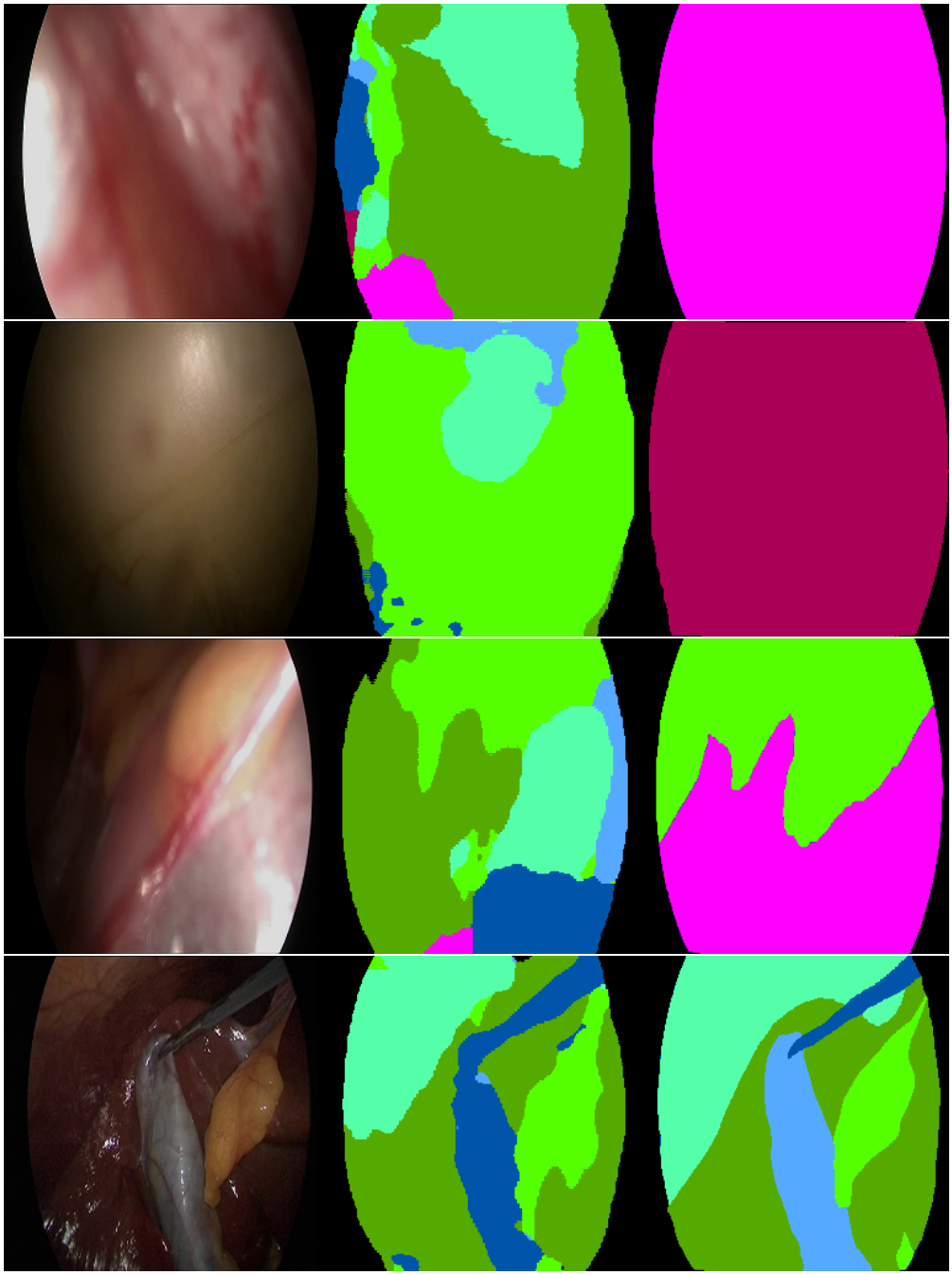

Figure 4.2: Predictions of our network on the miccaiSeg dataset with the single instrument class. The left column shows the original images, the middle column shows the prediction, while the right column shows the corresponding groundtruth.

some interesting directions in the future work section in Section 5.1 on how we can better leverage this information.

Class	IoU (Proposed Method)	IoU (UNet)	Precision (Proposed Method)	Precision (UNet)	Recall (Proposed Method)	Recall (UNet)	F1 Score (Proposed Method)	F1 Score (UNet)
Unknown	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
Instruments	0.73	0.51	0.79	0.68	0.91	0.67	0.85	0.68
Liver	0.77	0.54	0.84	0.61	0.90	0.84	0.87	0.70
Gallbladder	0.50	0.19	0.80	0.40	0.58	0.26	0.67	0.31
Fat	0.53	0.39	0.61	0.69	0.79	0.47	0.69	0.56
Upper Wall	0.41	0.08	0.65	0.15	0.53	0.14	0.58	0.14
Intestine	0.17	0.00	0.80	0.00	0.18	0.00	0.30	0.00
Artery	0.09	0.00	0.49	0.00	0.09	0.00	0.16	0.00
Black	0.94	0.90	0.96	0.92	0.97	0.94	0.97	0.93
Mean	0.46	0.29	0.66	0.38	0.55	0.37	0.57	0.37

Table 4.2: Comparison of the results of out network with U-Net [3], a popular architecture for Biomedical Image Segmentation.

4.2 All Categories

We additionally perform experiments for training and evaluation on all 19 classes of the miccaiSeg dataset. The results are shown in Table 4.3.

Again, we see similar trends as in the Single Instrument class case. The majority classes show good performance, especially Hook, Liver, Gallbladder, Fat. Some categories, however, are completely inaccurately predicted such as Unknown, Bipolar, Scissors, Irrigator, Trocars, Clip, Artery, Bile, and Blood. We can attribute the failure to the very few and often very difficult to discern instances of these classes in the training dataset.

Overall, we provide a baseline result in the problem domain. We will open source our work (code and dataset) to enable other researchers to contribute to the problem. We used the PyTorch Deep Learning framework [54] in our work, and our code is publically available at `https://github.com/salmanmaq/segmentationNetworks`.

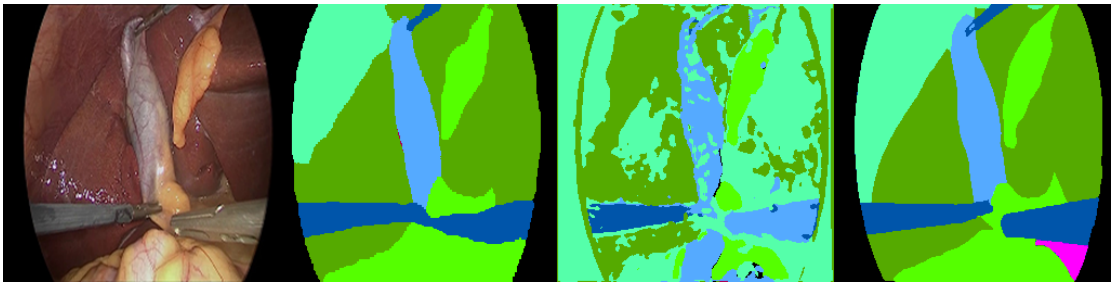

Figure 4.3: Sample of predictions of our network as compared to UNet [3], a popular CNN architecture for Biomedical image segmentation. The left column shows the input image, the 2nd column shows the prediction for our network, the 3rd column shows prediction with UNet, and the last column shows the groundtruth.

Class	IoU	Precision	Recall	F1 Score
Unknown	0.00	0.00	0.00	0.00
Grasper	0.39	0.64	0.49	0.56
Bipolar	0.00	0.00	0.00	0.00
Hook	0.64	0.70	0.89	0.78
Scissors	0.00	0.00	0.00	0.00
Clipper	0.34	0.51	0.51	0.51
Irrigator	0.01	0.71	0.01	0.02
Specimen Bag	0.22	0.50	0.29	0.37
Trocars	0.00	0.00	0.00	0.00
Clip	0.00	0.00	0.00	0.00
Liver	0.73	0.78	0.92	0.85
Gallbladder	0.53	0.71	0.68	0.70
Fat	0.55	0.69	0.73	0.71
Upper Wall	0.40	0.69	0.49	0.57
Intestine	0.20	0.64	0.22	0.33
Artery	0.00	0.01	0.00	0.00
Bile	0.00	0.00	0.00	0.00
Blood	0.00	0.00	0.00	0.00
Black	0.92	0.94	0.97	0.96
Mean	0.26	0.40	0.33	0.33

Table 4.3: Results of our network on the miccaiSeg with all the 19 classes.

Chapter 5

Conclusion

We introduced the problem of Laparoscopic Robotic Scene Segmentation in this thesis, introduced the miccaiSeg dataset, and proposed a baseline network to tackle the problem. We also showed that the unsupervised pre-training for Semantic Segmentation is beneficial for Semantic Segmentation as image reconstruction enables learning features similar to the features learnt for Semantic Segmentation.

We hope that this work paves way for future work in the domain as it extends the prior work done in Surgical scene understanding, and we propose it as a first step towards autonomous robotic surgeries. Below, we outline considerations for potential future work in this domain.

5.1 Future Work

Since surgical procedures are safety critical, it is essential that any proposed method is highly accurate. The same is true for other domains such as self-driving cars. We think that one of the biggest drawbacks of our approach was that some of the minority classes only have very few instances in our dataset. Especially classes like Clip, Blood, Bile, and Artery; which are also critical to detect, only have very few instances throughout the dataset. This is especially so because they only occupy very few pixels even in the images they are present in. Hence, the proposed miccaiSeg dataset should be extended so that there's a better class balance, and there are enough instances of the minority classes which would enable the learning discriminating features for those classes as well.

Secondly, as we say earlier that UNet [3] predictions retain the lower level information

such as edges and shapes. This shows that information sharing across the Encoder and Decoder networks can be useful. And that such an approach can also benefit from unsupervised pre-training. Secondly, powerful segmentation networks like DenseNet [48] can be trained if we have a larger dataset. Additionally, Dilated Convolutions [28] have been quite successful for Semantic Segmentation as they allow to capture context at multiple scales, thus providing a more global view of the input.

Recently, Pelt and Sethian [55] proposed a simple network which combines Dilated Convolutions with Dense blocks as in DenseNet. Here, they leverage the complementary properties of information sharing from lower layers to the later layers and information aggregation at multiple scales. Such an approach can be beneficial for our case as well.

Several works have pointed out that joint training for multiple objectives, including any auxiliary objectives can benefit the training stability and accuracy for all the tasks being trained for. This is because it enables the network to learn features which are more generic and suitable for multiple tasks across the dataset. Considering that, joint training for semantic segmentation along with tool presence detection can be beneficial. Similarly, considering temporal dependencies between successive video frames, temporal information can also be accounted for while making predictions for semantic segmentation. In this case, 3D CNNs [56] and Recurrent Neural Networks can be helpful for modeling the temporal dependencies.

Lastly, post-processing of Semantic Segmentation predictions due to spatial dependencies among neighboring pixels can be helpful to increase the network accuracy; but that usually comes at the cost of additional computational complexity and more complex network architectures.

References

[1] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30 (2):88–97, 2009.

[2] Cs231n: Convolutional neural networks for visual recognition. http://cs231n.github.io/convolutional-networks/. Accessed: 2018-02-20.

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

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32 (11):1231–1237, 2013.

[8] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22): 2402–2410, 2016.

[9] Ryan Poplin, Avinash V Varadarajan, Katy Blumer, Yun Liu, Michael V McConnell, Greg S Corrado, Lily Peng, and Dale R Webster. Predicting cardiovascular risk factors from retinal fundus photographs using deep learning. *arXiv preprint arXiv:1708.09843*, 2017.

[10] Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q Nelson, Greg S Corrado, et al. Detecting cancer metastases on gigapixel pathology images. *arXiv preprint arXiv:1703.02442*, 2017.

[11] Mikhail Volkov, Daniel A Hashimoto, Guy Rosman, Ozanan R Meireles, and Daniela Rus. Machine learning and coresets for automated real-time video segmentation of laparoscopic and robot-assisted surgery. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 754–759. IEEE, 2017.

[12] Ralf Stauder, Daniel Ostler, Michael Kranzfelder, Sebastian Koller, Hubertus Feußner, and Nassir Navab. The tum lapchole dataset for the m2cai 2016 workflow challenge. *arXiv preprint arXiv:1610.09278*, 2016.

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

[14] Yueming Jin, Qi Dou, Hao Chen, Lequan Yu, and Pheng-Ann Heng. Endorcn: recurrent convolutional networks for recognition of surgical workflow in cholecystectomy procedure video. *IEEE Trans. on Medical Imaging*, 2016.

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

[16] Andru P Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel De Mathelin, and Nicolas Padoy. Endonet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE transactions on medical imaging*, 36(1):86–97, 2017.

[17] Ashwin Raju, Sheng Wang, and Junzhou Huang. M2cai surgical tool detection challenge report. *University of Texas at Arlington, Tech. Rep*, 2016.

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

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[20] Christophe Doignon, Pierre Graebling, and Michel De Mathelin. Real-time segmentation of surgical instruments inside the abdominal cavity using a joint hue saturation color feature. *Real-Time Imaging*, 11(5-6):429–442, 2005.

[21] Luis C García-Peraza-Herrera, Wenqi Li, Caspar Gruijthuijsen, Alain Devreker, George Attilakos, Jan Deprest, Emmanuel Vander Poorten, Danail Stoyanov, Tom Vercauteren, and Sébastien Ourselin. Real-time segmentation of non-rigid surgical tools based on deep learning and tracking. In *International Workshop on Computer-Assisted and Robotic Endoscopy*, pages 84–95. Springer, 2016.

[22] Luis C García-Peraza-Herrera, Wenqi Li, Lucas Fidon, Caspar Gruijthuijsen, Alain Devreker, George Attilakos, Jan Deprest, Emmanuel Vander Poorten, Danail Stoyanov, Tom Vercauteren, et al. Toolnet: holistically-nested real-time segmentation of robotic surgical tools. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 5717–5722. IEEE, 2017.

[23] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[24] Wenqi Li, Guotai Wang, Lucas Fidon, Sebastien Ourselin, M Jorge Cardoso, and Tom Vercauteren. On the compactness, efficiency, and representation of 3d convo-

lutional networks: brain parcellation as a pretext task. In *International Conference on Information Processing in Medical Imaging*, pages 348–360. Springer, 2017.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[26] Mohamed Attia, Mohammed Hossny, Saeid Nahavandi, and Hamed Asadi. Surgical tool segmentation using a hybrid deep cnn-rnn auto encoder-decoder. In *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*, pages 3373–3378. IEEE, 2017.

[27] Daniil Pakhomov, Vittal Premachandran, Max Allan, Mahdi Azizian, and Nassir Navab. Deep residual learning for instrument segmentation in robotic surgery. *arXiv preprint arXiv:1703.08580*, 2017.

[28] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[29] Iro Laina, Nicola Rieke, Christian Rupprecht, Josué Page Vizcaíno, Abouzar Eslami, Federico Tombari, and Nassir Navab. Concurrent segmentation and localization for tracking of surgical instruments. In *International conference on medical image computing and computer-assisted intervention*, pages 664–672. Springer, 2017.

[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436, 2015.

[31] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[32] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[33] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[34] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[36] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[37] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[38] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision a pattern recognition*, pages 580–587, 2014.

[39] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

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

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[42] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[43] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[44] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[45] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[46] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[47] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015.

[48] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1175–1183. IEEE, 2017.

[49] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[50] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173, 2008.

[51] Hasan Sajid. Robust background subtraction for moving cameras and their applications in ego-vision systems. 2016.

[52] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

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

[54] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[55] Daniël M Pelt and James A Sethian. A mixed-scale dense convolutional neural network for image analysis. *Proceedings of the National Academy of Sciences*, page 201715832, 2017.

[56] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

Semantic Segmentation of Human Torso Region for Laparoscopic Surgery

ORIGINALITY REPORT

5%	3%	3%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to GLA University Student Paper	1%
2	"Neural Information Processing", Springer Nature, 2017 Publication	1%
3	"Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries", Springer Nature, 2018 Publication	<1%
4	campar.in.tum.de Internet Source	<1%
5	Mohamed Attia, Mohammed Hossny, Saeid Nahavandi, Hamed Asadi. "Surgical tool segmentation using a hybrid deep CNN-RNN auto encoder-decoder", 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2017 Publication	<1%
6	L. Merino, A. Ollero. "Computer vision techniques for fire monitoring using aerial images", IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02, 2002 Publication	<1%

7	Submitted to University College London Student Paper	<1%
8	"Computer Vision", Springer Nature, 2017 Publication	<1%
9	Mohammad Arifur Rahman, Nathan LaPierre, Huzefa Rangwala. "Phenotype Prediction from Metagenomic Data Using Clustering and Assembly with Multiple Instance Learning (CAMIL)", IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2018 Publication	<1%
10	C. Huang, J. R. G. Townshend. "A stepwise regression tree for nonlinear approximation: Applications to estimating subpixel land cover", International Journal of Remote Sensing, 2003 Publication	<1%
11	sse.tongji.edu.cn Internet Source	<1%
12	blog.cluster-text.com Internet Source	<1%
13	hal.archives-ouvertes.fr Internet Source	<1%
14	bmcbioinformatics.biomedcentral.com Internet Source	<1%
15	Luis C. Garcia-Peraza-Herrera, Wenqi Li, Lucas Fidon, Caspar Gruijthuijsen et al. "ToolNet: Holistically-nested real-time segmentation of robotic surgical tools", 2017 IEEE/RSJ International Conference on Intelligent Robots	<1%

	and Systems (IROS), 2017 Publication	
16	dione.lib.unipi.gr Internet Source	<1%
17	Kumar, S.. "An observation-constrained generative approach for probabilistic classification of image regions", Image and Vision Computing, 20030110 Publication	<1%
18	"Computer Vision – ECCV 2016", Springer Nature, 2016 Publication	<1%
19	bioimaging.med.yale.edu Internet Source	<1%
20	www.framkom.se Internet Source	<1%
21	Submitted to Campus 02 Fachhochschule der Wirtschaft GmbH Student Paper	<1%
22	www.d.umn.edu Internet Source	<1%
23	www.aanda.org Internet Source	<1%
24	tsukuba.repo.nii.ac.jp Internet Source	<1%
25	www.aclweb.org Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		