Learning to Extract Tumour Vasculature: Techniques in Machine Learning for Medical Image Analysis

Russell Bates

New College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2017

Abstract

Cancer is a leading cause of death worldwide with an estimated 14 million new cases occurring yearly and approximately 8 million deaths. Although much progress has been made in the understanding and treatment of cancer, there are still many mechanisms that remain poorly understood. The development of vasculature is known to be a key element in facilitating the growth of a tumour. Modern imaging modalities such as multi-photon fluorescence microscopy allow unprecedented opportunities to examine and quantify this vasculature *in vivo*. However, the appearance of vascular networks, imaged at these scales, can be extremely complex and the automatic delineation of such large, tortuous and chaotic vascular networks is a non-trivial task. In this thesis we develop a number of methods for the automatic delineation of tumour vasculature, imaged using *in vivo* microscopy. Recent developments in machine learning have provided a powerful set of techniques for the automated analysis of complex structures in images. Leveraging these, it is possible to develop algorithms, capable of learning from human annotations, which are able to analyse extremely large images quickly and with a high degree of accuracy. The key contributions of this thesis are as follows: we present a novel supervoxel algorithm for use in a lightweight machine learning framework for segmentation. We adapt the current state-of-the-art in segmentation using 2D deep fully convolutional neural networks for use in 3D vascular segmentation. We further demonstrate the use of hybrid convolutional-recurrent networks for extracting 3D vessel centrelines. We propose the use of Conditional Adversarial Networks for refining the extraction of vessel centrelines directly. Finally, we demonstrate the ability of the developed methods to make quantitative observations on longitudinal changes to *in vivo* tumour vasculature development.

Learning to Extract Tumour Vasculature: Techniques in Machine Learning for Medical Image Analysis



Russell Bates

New College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2017

Acknowledgements

Personal

I would firstly like to express my gratitude to my supervisors: Julia Schnabel, Vicente Grau and Veerle Kersemans for your guidance, advice and importantly, your willingness to let me follow my own instincts and research interests. I would like to thank all of my friends and colleagues at New College for providing such a wonderful home. It has been a joy to be a part of such a dynamic and inclusive MCR and it was a privilege to serve as Bar Rep. I would like to thank my colleagues in the IBME for providing a stimulating and exciting working environment, as well as those at KCL who welcomed me to my adopted home at St Thomas'. Specifically I would like to thank Ben Irving and Mike Brady, both of whose constant enthusiasm, knowledge and willingness to give advice were pivotal in shaping me as a researcher. I would like to thank Dan, Andrew and Amy for countless stimulating conversations (and arguments) at BigSoc, Tarifa and the Turf throughout the years. I would like to thank my family who have always encouraged me and who taught me the value of education and inquiry. And finally, I would like to thank Angharad, without whose love and support none of this would have been possible.

Institutional

This work would not have been possible without the generous funding of the EPSRC, Systems Biology, Doctoral Training Centre. I would like to thank the DTC and everyone involved, for being a source of support throughout my four years in Oxford. Not only did the DTC provide a warm welcome and a solid foundation for the start of my studies here, but it also dramatically broadened my horizons academically. Specifically, I would like to thank Prof. Elspeth Garman for her warmth and guidance throughout these four years. I would also like to thank my collaborators in the Department of Oncology, Bostjan Markelc in particular, for performing the experimental and imaging work without which, this project would not have existed.

Abstract

Cancer is a leading cause of death worldwide with an estimated 14 million new cases occurring yearly and approximately 8 million deaths. Although much progress has been made in the understanding and treatment of cancer, there are still many mechanisms that remain poorly understood. The development of vasculature is known to be a key element in facilitating the growth of a tumour. Modern imaging modalities such as multi-photon fluorescence microscopy allow unprecedented opportunities to examine and quantify this vasculature *in vivo*. However, the appearance of vascular networks, imaged at these scales, can be extremely complex and the automatic delineation of such large, tortuous and chaotic vascular networks is a non-trivial task. In this thesis we develop a number of methods for the automatic delineation of tumour vasculature, imaged using *in vivo* microscopy. Recent developments in machine learning have provided a powerful set of techniques for the automated analysis of complex structures in images. Leveraging these, it is possible to develop algorithms, capable of learning from human annotations, which are able to analyse extremely large images quickly and with a high degree of accuracy. The key contributions of this thesis are as follows: we present a novel supervoxel algorithm for use in a lightweight machine learning framework for segmentation. We adapt the current state-of-the-art in segmentation using 2D deep fully convolutional neural networks for use in 3D vascular segmentation. We further demonstrate the use of hybrid convolutional-recurrent networks for extracting 3D vessel centrelines. We propose the use of Conditional Adversarial Networks for refining the extraction of vessel centrelines directly. Finally, we demonstrate the ability of the developed methods to make quantitative observations on longitudinal changes to *in vivo* tumour vasculature development.

Contents

List of Figures **xi**

List of Abbreviations **xvii**

1 Introduction **1**

1.1 Thesis Contributions 5

1.2 Publications Arising From This Thesis 9

2 Background **11**

2.1 Data 12

2.1.1 Real Data 12

2.1.2 Synthetic data 16

2.1.3 Computational Resources 18

2.2 Vascular Analysis 19

2.2.1 Literature 22

2.3 Supervoxels 26

2.3.1 Literature 27

2.4 Graphical Models 29

2.4.1 Literature 32

2.5 Machine Learning 34

2.5.1 Decision Forests 34

2.5.2 Neural Networks 38

2.5.3 Deep Learning 43

2.5.4 Convolutional Neural Networks 48

2.5.5 Recurrent Networks 52

2.5.6 Adversarial Networks 55

2.5.7 Literature 57

2.6 Summary 60

3 Segmenting Vasculature with a Supervoxel-MRF and Random Forests 63
3.1 Introduction . . . 64
3.2 Related Work . . . 66
3.3 Materials . . . 68
3.3.1 Data . . . 68
3.4 Methods . . . 69
3.4.1 Supervoxels . . . 69
3.4.2 Markov Random Fields . . . 74
3.4.3 Random Forests . . . 79
3.4.4 Annotation . . . 80
3.5 Results . . . 81
3.6 Discussion . . . 87

4 Learning Vesselness with Convolutional Neural Networks 91
4.1 Method Outline . . . 92
4.1.1 Architecture . . . 93
4.1.2 Experiments . . . 94
4.2 Vesselness . . . 99
4.2.1 Vesselness as a Neural Network . . . 104
4.3 Model Combination . . . 106
4.4 Results . . . 108
4.5 Discussion . . . 113

5 Learning 3D Skeletons with Convolutional-Recurrent Networks 117
5.1 Introduction . . . 118
5.2 Related Work . . . 120
5.3 Materials . . . 121
5.4 Methods . . . 122
5.4.1 Architectures . . . 122
5.5 Results . . . 129
5.5.1 Results on Synthetic Data . . . 130
5.5.2 Results on Real Microscopy Data . . . 133
5.6 Discussion . . . 135

6 Refining 3D Skeletons with Conditional Generative Adversarial Networks 141
6.1 Introduction . . . 142
6.2 Methods . . . 144
6.3 Experiments . . . 148

6.3.1 Circle-Square Example 148
6.3.2 2D Vessel Example 152
6.3.3 Real Data 155
6.4 Discussion 159

7 Observations, Discussion and Outlook 163
7.1 Longitudinal Vessel Analysis 164
7.2 Summary 173
7.3 Discussion 175
7.4 Directions for Future Research 179
7.4.1 Adversarial Learning for Skeleton Extraction 179
7.4.2 Unify into a Multi-Task Learning Framework 180
7.4.3 Further Network Analysis 180

Appendices

A NetMets: Quantative Topological Measurements 185

B Statistical Testing 191
B.1 Wilcoxon Signed-Rank Test 191
B.2 Holm-Bonferroni Correction 192

C Skeletonisation Loss Functions 195

Bibliography 199

List of Figures

1.1 Fluorescence microscopy images of a whole tumour. Top Left: whole tumour, endothelium. Top Right: whole tumour, tumour cells. Bottom Left: single tile, endothelium. Bottom Right: single tile, tumour. 4

2.1 Different fluorescence channels for the *in vivo* multi-photon microscopy images. Top: Multi-channel overlay showing endothelium (cyan), tumour cells (green) and the perfusion marker (red). Bottom: Separated channels with endothelium (left), tumour cells (center) and perfusion (right). 14
2.2 3D renderings and 2D single slice images of the endothelium channel from the three different tumour types. 17
2.3 3D rendering of endothelium channel for two real data tiles (A and B). Corresponding derived skeletons are shown underneath (C and D). 18
2.4 (A) Real endothelium channel image, (B) Synthetic data image, (C) 3D rendering of synthetic segmentation, (D) 3D rendering of synthetic 3D skeleton. 19
2.5 Sum-of-angles-metric: the angles between sequential tangent vectors are summed. 20
2.6 Chord-length-ratio: The ratio between the chord connecting the two ends of the curve and the length of the curve is taken. 21
2.7 Recursively branching decision tree. Rectangles denote splitting nodes and ovals denote leaf nodes. 35
2.8 Different data points result in differing routes through the tree, ultimately arriving at different leaf nodes. 37
2.9 A simple neural network model with two hidden layers. 39
2.10 Network architecture schematic for the VGG-16 network (Simonyan and Zisserman, 2014). 50
2.11 Network architecture schematic for the U-Net network (Ronneberger et al., 2015). 52

3.1 Whole tumour images (225 tiles) showing endothelial cells (cyan) and the corresponding tumour cells (green). 65

3.2 Fluorescence channels showing a single tile (512 × 512 pixels) of endothelial cells (left) and the corresponding tumour cells (right). . 66
3.3 Top: Original noisy signal. Middle: Signal smoothed to eliminate noise (regular SLIC). Bottom: Signal transformed according to Eq. 3.3, recovering the hard edges. 72
3.4 Over-segmentation of vascular structure by two different supervoxel algorithms. Top: original SLIC algorithm, banding occurs along edges. Bottom: hesSLIC signal transform has significantly reduced banding. 72
3.5 hesSLIC as compared to monoSLIC and regular SLIC. Left: SLIC, center: monoSLIC, right: hesSLIC. 73
3.6 Transformation from endothelium channel (left) to ϕ domain (center) to superpixel segmentation (right). 74
3.7 Superpixels overlaid with estimated directional vectors v and magnitudes $f_{i,v}$, used in the calculation of G_4 in Equation 3.12. 77
3.8 Left: Annotation GUI implemented in MATLAB with sample multi-label annotation 'scribbles' shown. Right: Uncertainty overlaid using superpixels to help guide the annotator towards more informative labels. 81
3.9 Comparison of the effects of MRF strength on isotropic and anisotropic regularisation. 83
3.10 Left: Method of Narayanaswamy (Narayanaswamy et al., 2010). Right: Proposed Method. Here the method of Naranayaswamy struggles to fill the vessel lumina, leaving gaps in the segmentation. 84
3.11 Top left: Endothelial cell channel, top right: manual segmentation, bottom left: median filtering method, botton centre: vesselness filtering method, bottom right: proposed method. 84
3.12 Skeletons extracted from segmentation. Left: Proposed method, Middle: Ground Truth, Right: Median Filtering. 85

4.1 Top Left: Network trained to optimise a Dice based loss, corresponding BCE values are reported for each epoch. Top Right: Network trained to optimise a BCE based loss, corresponding Dice values are reported for each epoch. Bottom Left: Comparison of validation losses from both networks. Bottom Right: Comparison of Dice score at an optimal threshold for both networks. 97
4.2 Tightly packed vasculature, endothelium channel. 99
4.3 Small errors in segmentations may cause very large changes in derived centerlines. Segmentation shown in white and predicted centreline shown in red. 100

4.4 The original segmentation is shown in blue, and the resultant vesselness transform according to Equation 4.14 is shown in orange. The vesselness transform is visualised for a range of values of k. Adjusting this changes the specificity of the skeleton we would like to learn. . 101
4.5 Left: Slice through endothelium channel, Centre: manually annotated training data, Right: training data transformed according to Equation 4.14 with $k = 3$. 102
4.6 Image filtered to produce learned vesselness map. 103
4.7 Left: Endothelial channel, Center: Segmentation network, Right: Vesselness network. 103
4.8 Segmentation image multiplied by Vesselness image. 108
4.9 Left: Segmentation network, Right: Vesselness network. 108
4.10 Top: Segmentation network, Bottom: Vesselness network. 109
4.11 Box plots showing the comparison of the various methods under the Hausdorff average (max) measurement. 111
4.12 Comparison of Hausdorff Analysis including examples from Group 1 and Group 3 (top). Comparison between Hausdorff results from Group 1 and from Group 3. 112
4.13 Comparison of topology scores T_{max} and G_{max} for all methods. . . 113

5.1 Layer arrangement for a each depth layer of the U-Net architecture. 123
5.2 Network schematic for the equations given in Eq. 5.8. 125
5.3 Full combined architecture schematic for the U-Net2D + CLSTM networks. Image slices are passed independently through a shared CNN network, before being processed in parallel by stacked bidirectional ConvLSTM units. The final output will be approximately a 3D skeleton which can be refined to give our final skeletal representation. 126
5.4 Comparison of the U-Net2D method (left) with the proposed U-Net + ConvLSTM approach (right). A manually annotated ground truth is shown in blue (center). Detected vasculature is overlaid onto a volume rendering of the endothelium channel. 131
5.5 Distributions of Symmetric Hausdorff Average for the shallow and deep variants of the U-2D + CLSTM architectures. 134
5.6 Top: Distributions of Symmetric Hausdorff Average for the various competing methods (top left), we also show this distribution for the training set split into its component tumours (top right). Bottom: We show the p-values for a Wilcoxon signed-rank test between each competing method (bottom left). We also show whether this p-value corresponds to a significant difference (Wilcoxon Signed-Rank, $\alpha = 0.05$) after a Holm-Bonferonni correction (bottom right), with blue squares indicating significance. 135

5.7 Comparison of errors under the T_{max} (left) and G_{max} (right) metrics. 136
5.8 Network trained to reproduce segmentation (left), input endothelium channel (center) and network trained to reproduce skeleton (right). 138

6.1 Comparison of network schematics for supervised, generative adversarial and conditional generative adversarial networks. Top: Traditional supervised learning of the network G given training inputs x and training targets t. Middle: GAN training of the unsupervised training data t, seeded by random noise z. Bottom: Semi-supervised conditional GAN training of the network G given training inputs x and training targets t. 145
6.2 Examples of the input images overlaid with their respective target images. Input data is the large circle or square (white) and the training target is the small cross or square (green). 149
6.3 Qualitative comparison of the outputs from from the supervised and the semi-supervised adversarial networks. Input images are shown in white, training targets are shown in green and network predictions, given the respective input images are overlaid in red. 150
6.4 Two Synthetic vessel structures (white) with overlaid skeletons (green). Left: regular centreline. Right: with elastically deformed centreline. 151
6.5 Comparison of Hausdorff average between Adversarial training on elastic deformations (left), regular training on elastic deformations (centre) and regular training without deformations (right). 153
6.6 Comparison of the topological T_{max} score between Adversarial training on elastic deformations (left), regular training on elastic deformations (centre) and regular training without deformations (right). 154
6.7 Qualitative comparison of the network outputs for Adversarial (Elastic) and Regular (Elastic). We see that the thin skeleton structure has been learnt by the adversarial approach whereas the regular approach has learnt a more diffuse likelihood, incurring a break. . . 156
6.8 Comparison of Adversarial training (right) against purely supervised training (left) for learning vessel centrelines. The top and bottom rows show two different slices through the same image volume. . . . 157
6.9 Left: Comparison of the optimal symmetric Hausdorff average for the regular and adversarial training methods. Right: Comparison of the optimal T_{max} score, described in Section 5.5.2. 158
6.10 Traces for the symmetric Hausdorff average thresholded over the entire range shown for each test image. Each image in the training set is represented by a trace for each method. An average trace for each method is shown in bold. 159

7.1 Schematic for full segmentation and analysis pipeline. The segmentation model as described in Chapter 5 and the adversarially refined skeleton model as described in Chapters 5 and 6 are combined by taking a geometric average, as explored in Chapters 4 and 5. We estimate vessel diameters from the accurate segmentation model and extract the network topology from the skeletonised *mult 3d* image. The skeleton network is then traversed to form a discrete graphical representation of the network and features are attached to the edges and nodes of this graph. Analysis can then be efficiently performed over the entire vascular network. 165
7.2 Comparison of input image with segmentation and skeletonisation output. 168
7.3 Extracted feature graph of the entire tumour (top). Various local averages of features (bottom). 169
7.4 Changes in vessel branching density shown against days since treatment. Results are shown for the raw values (left) and the normalised values (right). Error bars show a 68% confidence interval for the mean. 170
7.5 Changes in vessel length (top) and diameter (bottom) shown against days since treatment. Results are shown for the raw values (left) and the normalised values (right). Error bars show a 68% confidence interval for the mean. 171
7.6 Left: Total number of vessels that exceed 100µm in diameter. Right: Proportion of vessels that exceed 100µm in diameter. 171
7.7 Comparison of the different tortuosity measurements Chord-Length Ratio (left) and Sum-of-angles-metric (right). 172
7.8 The progression of vessel extraction methods: Segmentation (left), Skeleton (centre) and Adversarial (right). 175
7.9 The progression of all methods proposed in the thesis (top) and the results for each method on each tumour category (right). 176

A.1 A hypothetical ROC curve shown in red. Here we show how the values T_{FPR}, T_{FNR} and T_{max} all relate to an upper and lower bound on the AUC for the unknown ROC curve. Lower and upper bounds are shown in yellow on the left and right figures respectively. 186
A.2 Example of the various graph constructions. Left: GT graph (blue) and T graph (green) with node assignment shown (red). Centre: GT graph (blue) is overlaid with the core connectivity graph GT_c (green). Right: T graph (blue) is overlaid with the core connectivity graph T_c (green). 189

List of Abbreviations

1D, 2D, 3D	One-, two- or three-dimensional, referring to the spatial dimensions in an image.
KLD	Kullback-Leibler Divergence
MSE	Mean Squared Error
BCE	Binary Cross-Entropy
MLE	Maximum Likelihood Estimator
NN	Neural Network.
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
FCN	Fully Convolutional Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
CLSTM	Convolutional Long Short-Term Memory
GAN	Generative Adversarial Network
CGAN	Conditional Generative Adversarial Network
DCGAN	De-Convolutional Generative Adversarial Network
WGAN	Wasserstein Generative Adversarial Network
VAE	Variational Auto-Encoder
MRI	Magnetic Resonance Imaging
DCE-MRI	Dynamic Contrast-Enhanced Magnetic Resonance Imaging
CT	X-ray Computed Tomography
GT	Ground Truth
MAP	Maximum a Posteriori
RF	Random Forest
SVM	Support Vector Machine

PCA . . . . . Principal Components Analysis

SOAM . . . . . Sum-Of-Angles-Metric

CLR . . . . . Chord-Length-Ratio

SLIC . . . . . Simple Linear Iterative Clustering

1 Introduction

As tumours grow, it becomes impossible for them to acquire sufficient oxygen and nutrients to facilitate growth by diffusion alone. A distance of 0.2mm is considered to be the maximum distance from a blood vessel for a tumour cell to maintain growth (Weinberg, 2013). In order to address this, the tumour must cultivate its own vascular system, so that it may maintain sufficient levels of perfusion. The formulation of vasculature during embryonic development is genetically programmed and, as such, forms well organised and efficient vascular networks. Tumours build up their vasculature in a number of ways, both by recruiting from the surrounding vascular bed as well as encouraging the production of neovasculature via the signalling molecule VEGF, seeded by circulating Endothelial Progenitor Cells (EPCs). This vasculature is formed as a result of local interactions and, as such, lacks an overarching structure. This leads to the archetypal 'chaotic vasculature' of the tumour, with highly tortuous vessels that are characteristic of tumour vasculature and exhibit higher levels of permeability than regular vasculature (Danhier, 2016). The chaotic assembly of endothelial cells in the formation of this neovasculature can leave gaps up to several microns wide, and the capillaries have been estimated to be up to 10 times more permeable than normal capillaries (Weinberg, 2013). This ability to induce angiogenesis was listed as one of Hanahan and Weinberg's *Hallmarks of Cancer* (Hanahan and Weinberg, 2011) and a full understanding of how the vasculature grows and remodels itself is thought to be key to a full understanding of cancer.

As the formation of vasculature appears to play such an important role in the development and growth of certain types of tumours, it is not surprising that it has been targeted as a potential approach to therapy. The emergence of the monoclonal antibody Bevacizumab demonstrated the efficacy of anti-angiogenic treatments in humans. Bevacizumab targets the pathway inducing vascular growth by binding to the angiogenic growth factor VEGF. Although expectations have been moderated slightly since their arrival (Bergers and Hanahan, 2008), they still remain a valuable tool in the clinical treatment of cancer. Another class of vascular targeting drugs are the Vascular Disrupting Agents (VDAs). The

VDA Compretastatin A-4, whose efficacy is currently being investigated for use in treatment of a number of different cancers, is thought to work by disrupting the actin and tubulin cyto-skeletons, leading to vascular shut-down (Tozer et al., 2005). However, the full mechanism for this still remains unclear.

It has been hypothesised by Jain (Jain, 2001) that the efficacy of anti-angiogenic therapies in combination with radio-/chemo-therapy is due to the re-normalising effect the anti-angiogenic drugs have on the normally chaotic tumour vasculature. The environment created by the poor vasculature in tumours is hypoxic, with high interstitial pressures. This makes it extremely difficult to deliver cytotoxic drugs to the tumour itself. Jain suggests that the anti-angiogenic therapy may 'prune' the more disordered vessels leaving a more 'normal' vasculature. The elimination of excess endothelial cells has also been shown to reduce interstitial pressure, further normalising the micro-environment and improving drug delivery. Clearly, the effects of vascular normalisation at this scale are not yet fully understood, and the ability to properly visualise the effects at this scale are only now being realised. The examination of more mature tumour vasculature, at larger scales, is also of great interest. Vascular perfusion, angiogenesis and the response to therapy have been investigated in the whole tumour using Dynamic Contrast-Enhanced Magnetic Resonance Imaging (DCE-MRI) (Jackson et al., 2007; Connor et al., 2011). However, DCE-MRI only provides us with an indirect view of perfusion via leakage, rather than a true view of the vasculature.

In pre-clinical research, the introduction of contrast enhanced micro-CT is allows highly detailed views of the vasculature directly (Kersemans et al., 2015). Ehling *et al.* (Ehling et al., 2014) have shown promising results for the quantitative study of *in vivo* micro-CT by correlating their *in vivo* results with high-resolution *ex vivo* imaging as well as immunohistology stainings. Work on the analysis of the structure of large scale tumour vasculature by Bullitt *et al.* (Bullitt et al., 2003) examined various metrics for the analysis of tortuous vasculature. One of their key contributions was the development of a discriminative measure of tortuosity in the *sum-of-angles* metric (SOAM), showing that it is able to capture the characteristic 'wiggliness' of

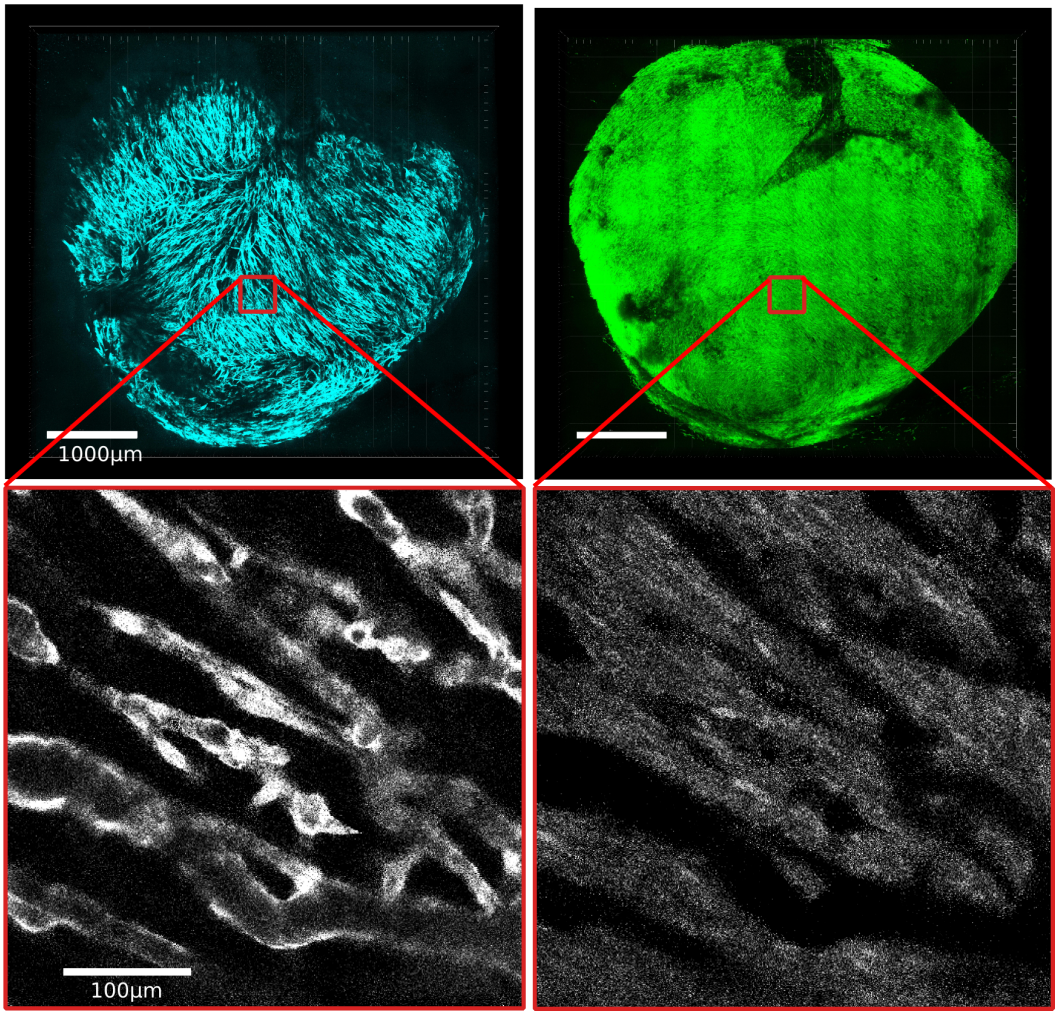


Figure 1.1: Fluorescence microscopy images of a whole tumour. Top Left: whole tumour, endothelium. Top Right: whole tumour, tumour cells. Bottom Left: single tile, endothelium. Bottom Right: single tile, tumour.

tumour vasculature. This metric was then used in later work to show a correlation between vascular tortuosity and tumour malignancy (Bullitt et al., 2005).

The majority of pre-clinical, *in vivo*, vascular imaging modalities rely on the injection of a contrast agent to enhance the vasculature, however, this may be problematic for tumour imaging. The leaky and irregularly perfused nature of the vasculature formed during angiogenesis, mean that a bloodpool labelling may not be sufficiently reliable. Fluorescence microscopy provides an alternate approach to bloodpool labelling by fluorescently tagging the endothelial cells. Whereas each section of the vasculature may or may not be well perfused at the time of imaging, each section of vasculature comprises many endothelial cells. By imaging

the endothelium directly it is possible to achieve a much more reliable signal and one that is much less biased towards well perfused sections of vasculature. Through the use of abdominal window chamber models, intra-vital imaging is made possible (Pittet et al., 2011), allowing extremely high resolution images of living tissue. Where the micro-CT images can acquire images at approximately 30μm isotropically, the intra-vital fluorescence microscopy is able to achieve an anisotropic voxel-size of 0.8μm × 0.8μm × 5μm with a theoretical in-plane resolution of 0.4μm and an axial resolution of 1.3μm. This makes it possible to image vascular formation at the lowest level, allowing an analysis and quantification of the angiogenic process. An example of the level of detail made possible by imaging of this kind is shown in Figure 1.1. Imaging of this kind is not possible in a clinical setting and for this reason, the work in this thesis will focus entirely on developing methods to assist pre-clinical imaging studies.

In order to extract quantitative measures from vasculature, it is necessary to first perform some kind of 'delineation' of the vasculature. For many experimentalists examining vasculature through microscopy imaging, manual or semi-automatic approaches are still the norm (Meijer et al., 2016; Bentley et al., 2014; Huang et al., 2014). Although manual delineations may be feasible for smaller data sets, to enable analysis of large amounts of vasculature over a range of tumours it is vital that efficient automated methods exist.

1.1 Thesis Contributions

In this thesis we will present a number of methods to perform the automatic delineation of tumour vasculature in large, 3D microscopy image volumes of tumour vasculature, *in vivo*. This is an extremely challenging task which poses a great number of difficulties when compared to more standard vascular segmentation tasks and innovation is required at each stage. After presenting our developed pipeline for extraction, we will finish with some observations enabled by the preceding methods. This thesis is structured as follows:

- Chapter 2: An introduction to the key concepts required to apply machine learning to vascular segmentation. We will briefly introduce the broad field of vascular segmentation before introducing the technical advances that have enabled the machine learning techniques upon which we will build in this thesis. In particular we will discuss the recent advances in deep learning. We will also introduce the relevant state-of-the-art literature for the methods discussed, as well as identifying the shortcomings for this particular task.

- Chapter 3: We propose a method for segmentation of the vasculature, combining superpixels, Random Forests and Markov Random Fields (MRF). This method allows efficient training from sparsely annotated samples. In this chapter we propose a novel supervoxel clustering algorithm called hesSLIC. We also propose a principled MRF regularisation technique, over the supervoxels, that incorporates vesselness and directionality.

- Chapter 4: We demonstrate the ability to use state-of-the-art methods from deep learning for the segmentation of our 3D vasculature. We also propose a method for the segmentation of the vasculature specifically, by approximating a 3D 'vesselness' function. Here we explore how 2D Convolutional Neural Networks (CNNs) can be used to learn a 3D structure such as a vesselness function.

- Chapter 5: We propose an architecture combining convolutional and recurrent networks to learn directly from the 3D vessel centrelines. Here we show that by combining 2D CNNs, using a Convolutional-Recurrent unit, we are able to learn much more complex 3D structures with minimal additional model complexity.

- Chapter 6: We propose the use of recent advances in adversarial training to refine 'vesselness' learning into direct skeleton extraction. Here we show how adversarial learning can be used to compensate for uncertainty and subjectivity in annotations and allow a more confident prediction of the vessel

centreline. We also show that this refinement reduces our previous dependence on choosing an 'optimal threshold'.

- Chapter 7: We present some observations enabled using the preceding methods. We discuss the findings of this thesis and present a number of directions for future research.

We briefly summarise the key contributions of this thesis as follows:

- We propose a novel supervoxel algorithm, aimed to address the issues in applying supervoxel clustering algorithms to medical images.
- We propose a vessel-oriented MRF regularisation scheme for segmentation via supervoxel labeling.
- We demonstrate the ability for deep convolutional neural networks to segment vasculature in multi-photon, *in vivo*, tumour microscopy volumes.
- We propose a training scheme to approximate a 3D 'vesselness' function using 2D CNNs. We show that this training scheme can improve the quality of vessel delineation both geometrically and topologically.
- We propose the use of Convolutional LSTMs to extract 3D vessel structures from multi-photon, *in vivo*, tumour microscopy volumes. We show that a minimal extension to a CNN architecture allows us to learn complex 3D structures directly.
- We demonstrate the ability to learn discrete skeletonisation of structures directly, using Conditional Generative Adversarial Networks. We show that learning centrelines in this way reduces a dependency on thresholding, and may improve the topological quality of the delineations.
- We show that through the automatic delineation of tumour vasculature, we are able to quantify distinct changes in vessel shape and distribution in response to two different types of therapy.

Although we had initially intended for the analysis of the experimental work to play a larger role in this thesis, delays in the experimental pipeline and an increased focus on the methodological contributions have prevented this. Instead, we present some preliminary experiments in Chapter 7 and suggest a number of avenues for further investigation in this area.

1.2 Publications Arising From This Thesis

Work that has contributed to this thesis, or applied ideas developed in this thesis may also be found in the following publications:

- *Refining the extraction of skeletal structures using Conditional Generative Adversarial Networks*, **R Bates**, B Irving, B Markelc, J Kaeppler, R Muschel, V Grau, JA Schnabel. *(In Preparation)*
- *Extracting 3D Vascular Structures from Microscopy Images using Convolutional Recurrent Networks*, **R Bates**, B Irving, B Markelc, J Kaeppler, R Muschel, V Grau, JA Schnabel. *(Under Review, arXiv preprint arXiv:1705.09597)*
- *Segmentation of Vasculature from Fluorescently Labeled Endothelial Cells in Multi-Photon Microscopy Images*, **R Bates**, B Irving, B Markelc, J Kaeppler, R Muschel, V Grau, JA Schnabel. *IEEE, Transactions on Medical Imaging, 2017*
- *Estimating oxygen distribution from vasculature in three-dimensional tumour tissue*, DR Grimes, P Kannan, DR Warren, B Markelc, **R Bates**, RJ Muschel, M Partridge. *Journal of The Royal Society Interface, 2016*
- *Filling large discontinuities in 3D vascular networks using skeleton-and intensity-based information*, **R Bates**, L Risser, B Irving, BW Papież, P Kannan, V Kersemans, JA Schnabel. *International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015*
- *Improving in vivo high-resolution CT imaging of the tumour vasculature in xenograft mouse models through reduction of motion and bone-streak artefacts*, V Kersemans, P Kannan, JS Beech, **R Bates**, B Irving, S Gilchrist, PD Allen, J Thompson, P Kinchesh, C Casteleyn, JA Schnabel, M Partridge, RJ Muschel, SC Smart. *PloS one, 2015*

2 Background

Contents

2.1 Data 12
2.1.1 Real Data 12
2.1.2 Synthetic data 16
2.1.3 Computational Resources 18
2.2 Vascular Analysis 19
2.2.1 Literature 22
2.3 Supervoxels 26
2.3.1 Literature 27
2.4 Graphical Models 29
2.4.1 Literature 32
2.5 Machine Learning 34
2.5.1 Decision Forests 34
2.5.2 Neural Networks 38
2.5.3 Deep Learning 43
2.5.4 Convolutional Neural Networks 48
2.5.5 Recurrent Networks 52
2.5.6 Adversarial Networks 55
2.5.7 Literature 57
2.6 Summary 60

2.1 Data

We will begin by outlining the kind of data used for all experiments within this thesis, the acquisition procedures for the data and a discussion of the details and complications arising from this data. For the demonstration and evaluation of the methods developed here we use both real, *in vivo*, imaging of tumour vasculature as well as synthetic data. While it is vitally important to test the methods on the real application data, synthetic data gives us a better handle on an exact ground truth, which can be difficult to acquire, both due to the time taken to create manual annotations and often the complex nature of the images themselves, especially when working in 3D. Using synthetic data when evaluating components of a pipeline can also help us prevent from over-fitting to the real data set.

2.1.1 Real Data

For our experiments on real data, we have delineated vasculature from images acquired using high resolution, multi-photon, fluorescence microscopy. This imaging approach achieves a theoretical lateral resolution of 0.4µm and an axial resolution of 1.3µm. Voxels are sized 5µm in the z direction and 0.83µm $\times$ 0.83µm in the x-y plane. Imaging was performed using an abdominal window chamber model in mice. This allows for intra-vital imaging of the tumours (Ritsma et al., 2013). The abdominal window chamber was surgically implanted in transgenic mice on C57Bl/6 background that had expression of red fluorescent protein tdTomato in the endothelial cells only. The murine colon adenocarcinoma MC38 tumours with expression of green fluorescent protein (GFP) in the cytoplasm were induced by injecting 5µm of dense cell suspension in a 50/50 mixture of saline and matrigel (Corning, NY, USA). The images of tumours were acquired 9 - 14 days after tumour induction with Zeiss LSM 880 microscope (Carl Zeiss AG), connected to a Mai-Tai tunable laser (Newport Spectra Physics). An excitation wavelength of 940 nm was used and the emitted light was collected with Gallium Arsenide Phosphide (GaAsP) detectors through a 524 – 546 nm bandpass filter for GFP and a 562.5 – 587.5 nm bandpass filter for tdTomato and with a multi-alkali PMT detector through a 670 –

760 bandpass filter for Qtracker® 705. A 20× water immersion objective with a numerical aperture of 1.0 was used to acquire a Zstacks-TileScan with dimensions of 512×512 pixels in x and y, and approximately 70 planes in z, with a z step of 5µm. Each tumour is covered by approximately 100-200 tiles, depending on the size. We define a 'tile' to mean a small subset of the $x - y$ extent of a larger image, composed of a number of slices in the z direction. A 'slice' therefore refers to a single two dimensional image, a stack of which composes a tile. All animal studies were performed in accordance with the Animals Scientific Procedures Act of 1986 (UK) and Committee on the Ethics of Animal Experiments of the University of Oxford. All experimental work has been carried out by Bostjan Markelc and Jakob Kaeppler in the lab of Ruth Muschel, within the Oxford Institute for Radiation Oncology, Department of Oncology, University of Oxford.

By using both a labelled blood-pool based agent (Qtracker®705) and a transgenic mouse model with fluorescently labelled endothelium, we are able to separate the related but distinct quantities of the presence of vasculature and the perfusion of said vasculature. Although in healthy vasculature, perfusion is commonly used as a proxy for imaging the vasculature itself, this can not always be relied upon in the vasculature found within tumours.

For the purposes of manual annotation and experimentation we take a number of smaller tiles from each large tumour image volume, containing approximately $512 \times 512 \times 70$ voxels and three fluorescence channels, corresponding to endothelium, tumour cells and perfusion levels. An example of these three channels is shown in Figure 2.1. It is clear to see from these images that the perfusion marker, while clearly highlighting the larger vessels, does not reliably enhance the smaller vessels present. The endothelium is very clear, everywhere that it is present, however we run into the issue that the vessel lumen is not imaged in this channel. Information about the larger vessel lumina must be inferred from both the context within the endothelium channel and from the absence of signal in the tumour channel as the presence of a vessel in these regions naturally excludes the presence of tumour cells.

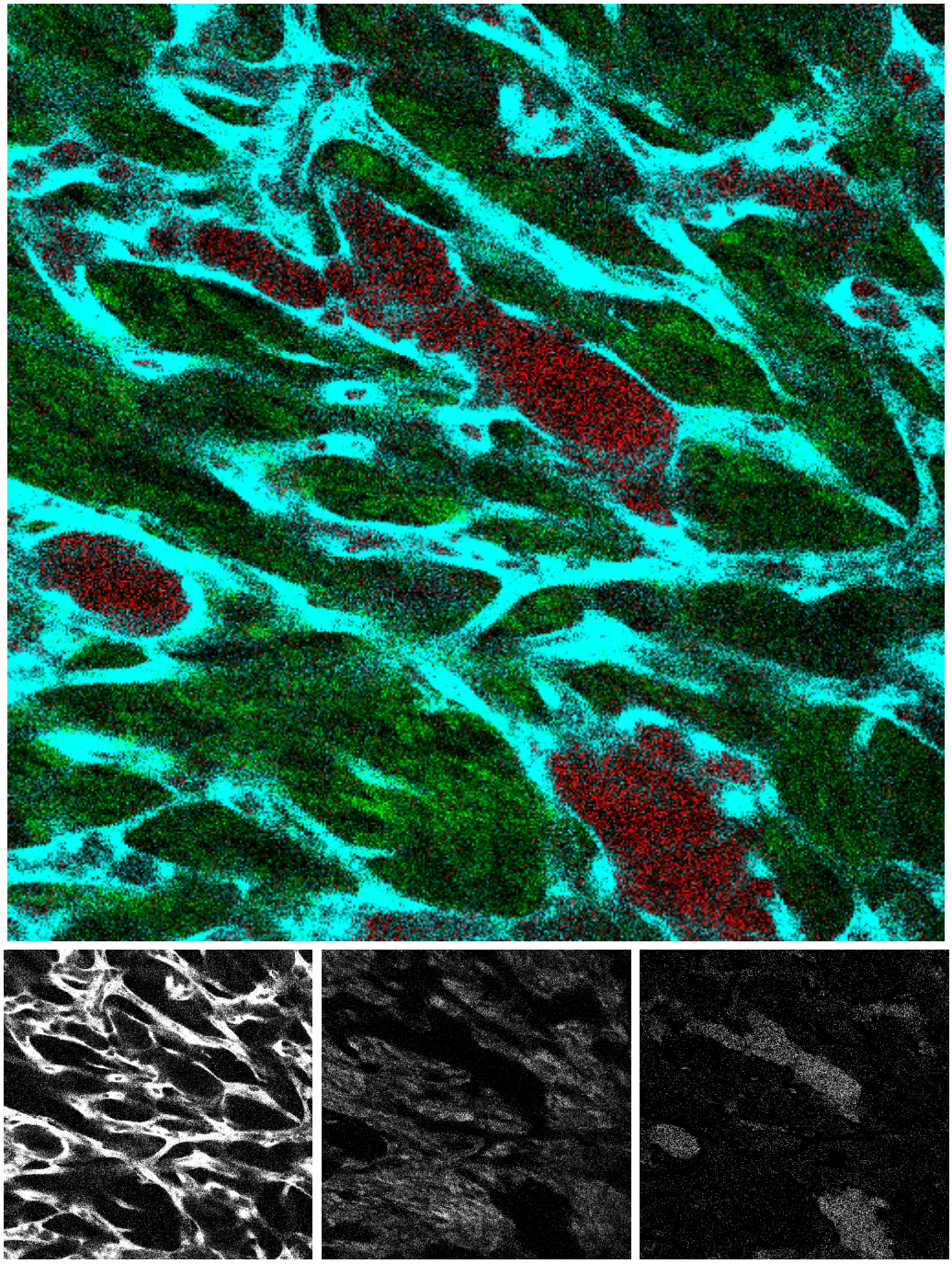


Figure 2.1: Different fluorescence channels for the *in vivo* multi-photon microscopy images. Top: Multi-channel overlay showing endothelium (cyan), tumour cells (green) and the perfusion marker (red). Bottom: Separated channels with endothelium (left), tumour cells (center) and perfusion (right).

As has been discussed in our motivation, we are interested in segmenting the vasculature present in these images. However, beyond that, we are also interested in extracting the topology and connectivity of the network as a whole. In order to do this the segmentation must be reduced to some medial representation that defines both the geometry and the topology of the network, such as a skeleton. We must therefore produce ground truth versions of both the segmentation and of medial representation skeletons. We define 'skeletons' as a binary image representation of the parent image, such that each pixel has, at most, two neighbouring non-zero valued pixels, with the exception of 'branch points' which may have more neighbours. The 'end points' of a skeleton will have only one non-zero neighbour. Here we define the set of possible neighbours as the full, 26-voxel neighbourhood. This skeletal representation provides us with an explicit (locally) one dimensional representation of the vasculature. This allows us to parametrise the vessel and compute and assign features in a way which is abstracted from the appearance of the vasculature itself.

In order to produce the manual segmentations we utilised the ITK-SNAP[1] software application for 3D medical image processing. This process is fully manual and does not rely on any semi-automated approaches. All annotations used for the training and validation of models were produced by an independent annotator with domain specific expertise. Skeletons are derived by first producing a manual segmentation of the vasculature on each slice. This is then followed by a centreline extraction of this manual segmentation in 3D, using the NeuTube[2] software package (Feng et al., 2015). Extracted skeletons are then manually pruned and refined using the NeuTube software package. Examples of these derived skeletons are shown in Figure 2.3. The training dataset consists of 25 manually segmented tiles of size $512 \times 512 \times 70$, which are then subdivided, with overlap, to form a larger training set of smaller image stacks. We hold back 10% of our training data for validation and hyper-parameter tuning. The tiles were taken from 3 different tumour categories:

1. Group 1: A control group of the MC38 tumour cell line.

[1] http://www.itksnap.org
[2] http://www.neutracing.com/

2. Group 2: MC38 tumours treated with DC101 antibodies, which reduce VEGF signalling and should cause more 'normal' vasculature.

3. Group 3: MC38 tumours treated with Anti-Dll4 antibodies. Blocking of Dll4 signalling causes 'hyper-sprouting' behaviour and creates very dense, complex vascular networks.

Examples of the endothelial channel from all three different tumour categories are shown in Figure 2.2.

In addition to being highly anisotropic, there are a number of other issues associated with live animal imaging which must be considered by our methods. The primary issue encountered here is the slice misalignment induced by animal breathing motion. As the images are acquired in a tile-scan, there will be a non-trivial time delay between the acquisition of subsequent slices in each tile z-stack and between slices in neighbouring tiles. This time delay can result in uncorrelated motion between nearby slices, which may cause problems when performing segmentation and other processing steps.

2.1.2 Synthetic data

In order to test our method against a known ground truth, we have also generated a synthetic dataset that presents the same issues as our real data. The synthetic data consists of a number of hollow, tubular structures that are tightly packed and represented with anisotropic voxel sizes, comparable to the real data. The vessels are generated by first iteratively growing the centrelines throughout the volume. Then the segmentation mask is produced by dilating this skeleton to some randomly selected radius (chosen to reflect the range of sizes visible in the real data). We then generate two distance maps d_1 and d_2 that represent the distance from the nearest foreground and nearest background respectively. The endothelium is then generated from these distance maps using the following formula:

$$E = \exp(-d_1/\sigma_1)\exp(-d_2/\sigma_2), \tag{2.1}$$

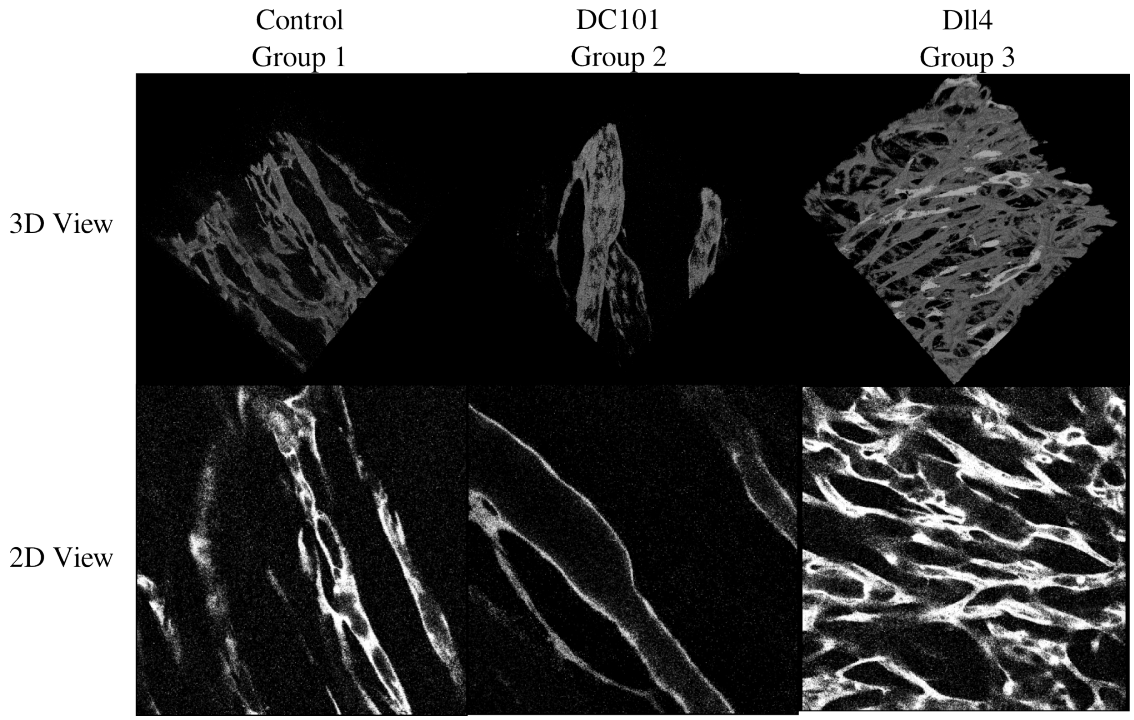


Figure 2.2: 3D renderings and 2D single slice images of the endothelium channel from the three different tumour types.

where σ_1 and σ_2 are tuned to give a qualitative appearance similar to our data. This allows us to tune a decay in signal away from the highly visible endothelium comprising the vessel wall. A steep drop off externally is governed by a small value for σ_1 and a more gradual drop off of signal internally to the vessel is governed by a larger σ_2. The synthesised image volume is re-sampled anisotropically to the same aspect ratio as the real data.

Due to the anisotropy, many intersections between vessels are visible and the algorithm must reconstruct the 3D branching structure. We add Gaussian noise, whose variance is tuned to the background noise in the real images. We also add 'Salt and Pepper' and Poisson noise to simulate the detector noise present in fluorescence microscopy. Slice 'jitter' is also added to simulate the slight slice misalignment present in the real data, due to mouse breathing. We use a uniformly sampled jitter in the range $[-5, 5]$ voxels in both the x and y direction. Examples of this data are shown in Figure 2.4.

For the synthetic experiments we train on 20 image volumes, with size $512 \times$

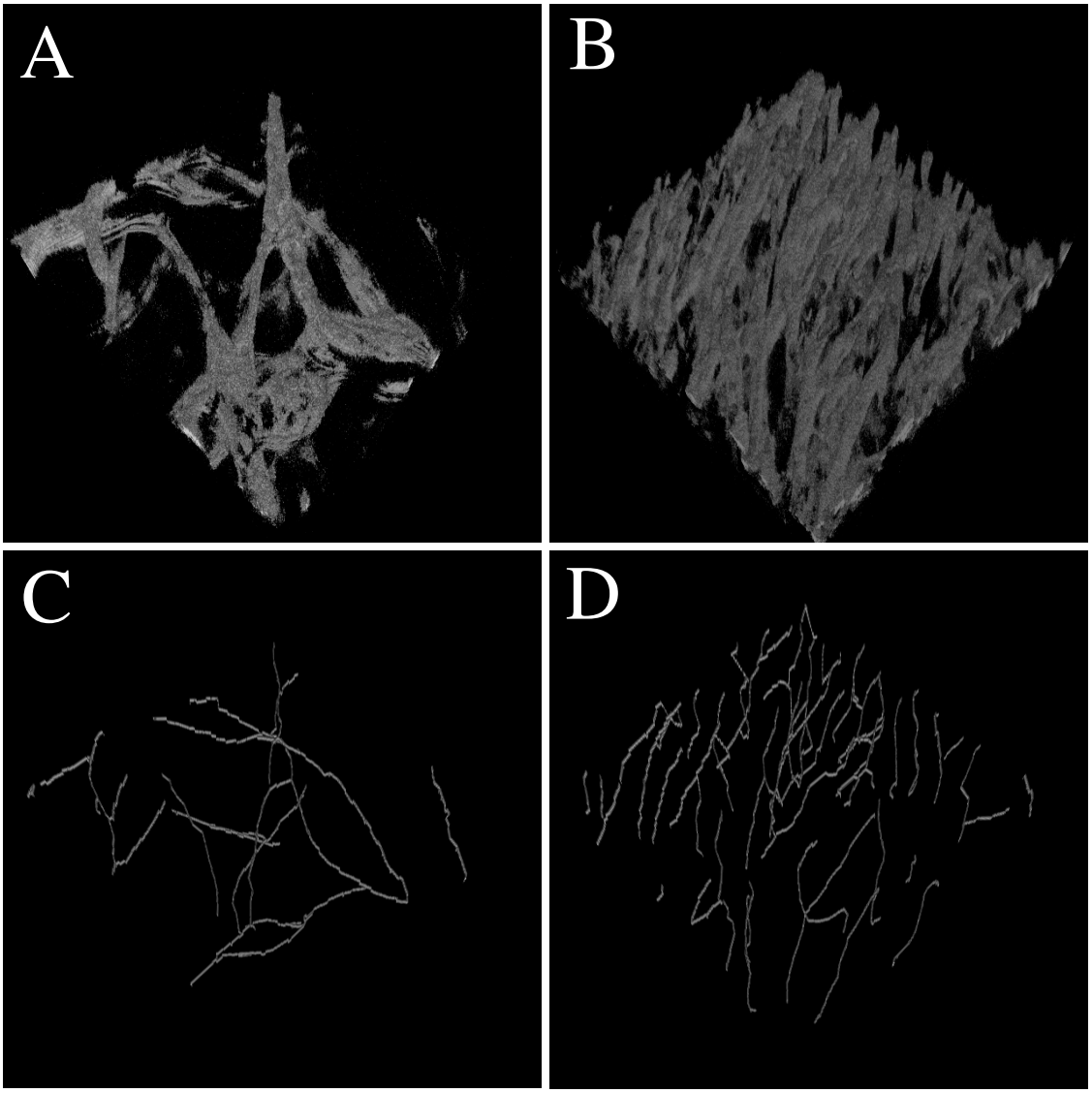


Figure 2.3: 3D rendering of endothelium channel for two real data tiles (A and B). Corresponding derived skeletons are shown underneath (C and D).

512×40, generated in this manner.

2.1.3 Computational Resources

All computation for Chapter 3 is performed using MATLAB. All computation for the remaining chapters is performed in Python, using the TensorFlow[3] and Keras[4] libraries for the purposes of training and executing deep neural networks. All training of deep neural networks is performed using NVIDIA Tesla K40m 12GB

[3]https://www.tensorflow.org/
[4]https://keras.io/

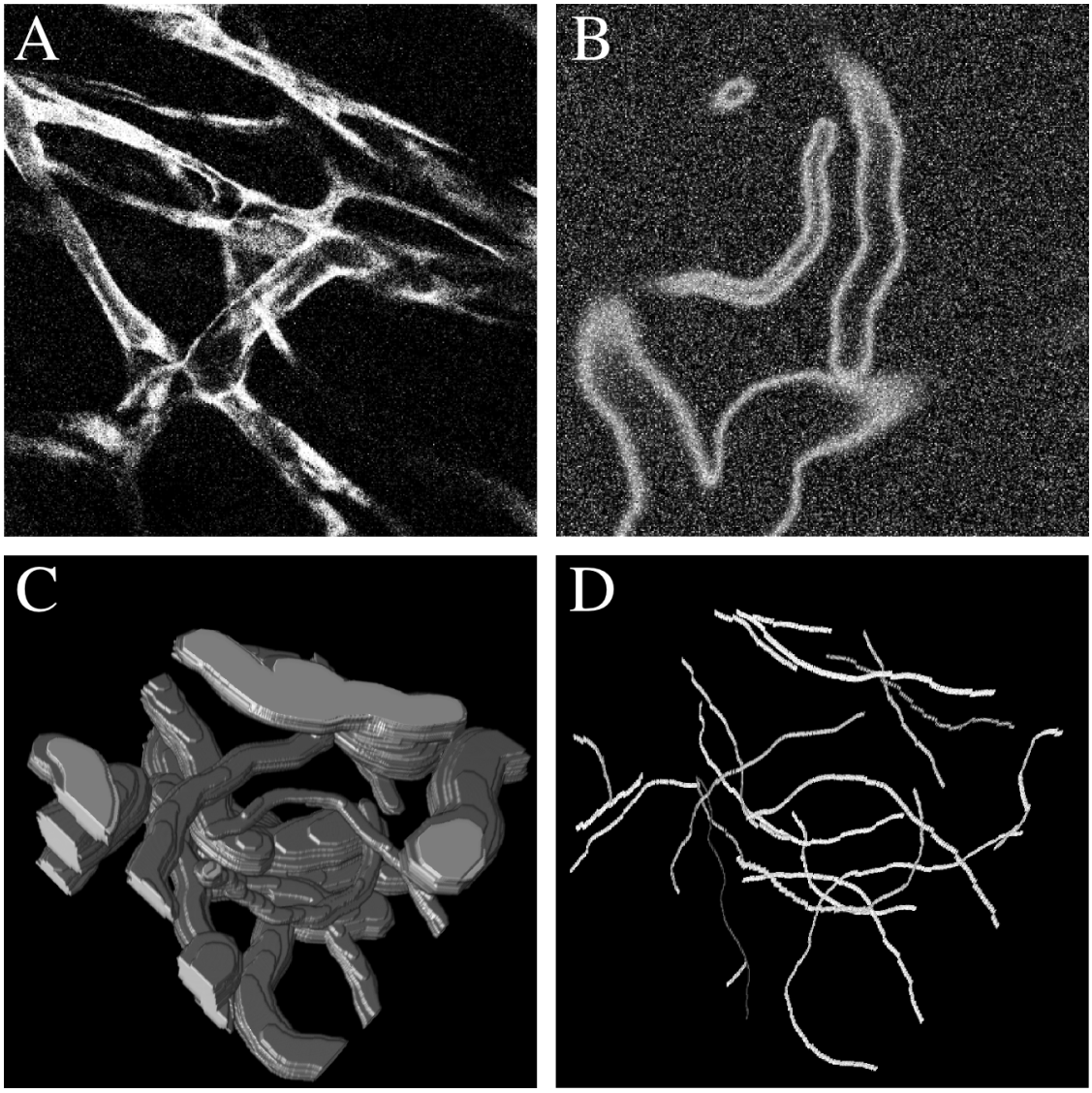


Figure 2.4: (A) Real endothelium channel image, (B) Synthetic data image, (C) 3D rendering of synthetic segmentation, (D) 3D rendering of synthetic 3D skeleton.

GPUs. Additionally, we would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work[5].

2.2 Vascular Analysis

The nature and role of vasculature mean that any numerical characterisations may examine both topological and morphological properties. The local appearance of vasculature can be quantified by measurements such as diameter and tortuosity.

[5]http://dx.doi.org/10.5281/zenodo.22558

However, vasculature importantly exists as a network, and as such we must also be mindful of the topological, properties of the network as a whole. This includes measurements such as branching length, branching order, lacunarity, fractal dimension and vessel density (Gould et al., 2011). Measurements of tortuosity in tumour associated vasculature were used by Bullitt *et al.* (Bullitt et al., 2003, 2005) to assess the malignancy of brain tumours. In doing this, they assess the merits of a number of different measurements of tortuosity, which is a subjective term. The method they propose is the Sum-of-angles-metric (SOAM), the sum of angles between regularly sampled tangent vectors along the centreline of a vessel, this is demonstrated in Figure 2.5. This will be zero for straight vessels and may increase indefinitely for highly tortuous vessels. Since this metric will naturally increase monotonically with length, they also consider a length normalised version.

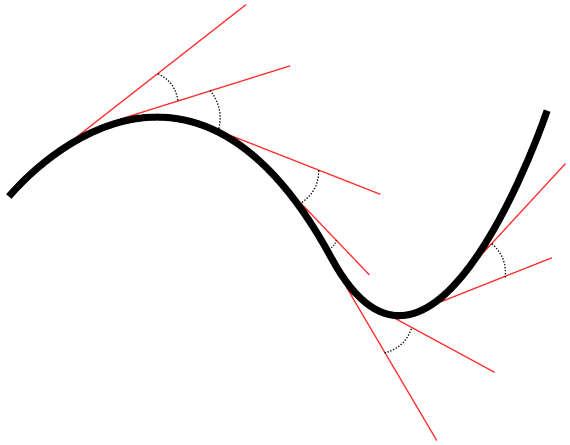


Figure 2.5: Sum-of-angles-metric: the angles between sequential tangent vectors are summed.

Another possible measure of tortuosity is the Chord-length-ratio, the ratio between the distance between the end points (the chord) and the length of the vessel. This will be approximately 1 for a straight vessel and will tend to zero for highly curved vessels. This measurement is demonstrated in Figure 2.6.

Although the topic of vascular segmentation has a large overlap with the general segmentation techniques applied across different applications, there are a number of features that set it apart. The core observation comes from the morphology of vasculature, in that it most commonly has a locally cylindrical appearance. This

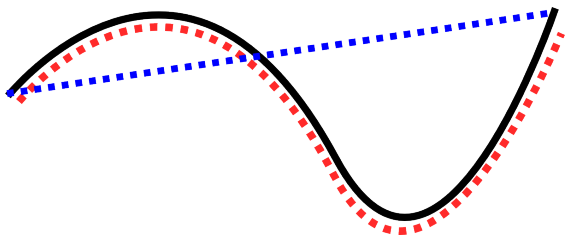


Figure 2.6: Chord-length-ratio: The ratio between the chord connecting the two ends of the curve and the length of the curve is taken.

observation forms the core of vessel segmentation techniques. The foundational works in exploiting this appearance model are those of Frangi *et al.* (Frangi et al., 1998), Sato *et al.* (Sato et al., 1997), Lorenz *et al.* (Lorenz et al., 1997) and Krissian *et al.* (Krissian et al., 1998). In all of these methods, it is observed that a tubular appearance can be modelled locally by an examination of the local Hessian structure, taken at a suitable scale. The eigenvalues of the Hessian matrix can be taken to represent the principal curvatures in the direction of the eigenvectors, which, since the Hessian is a symmetric matrix, form an orthonormal basis. In short, these papers conclude that a tubular region may be enhanced by searching for regions which possess a single small eigenvalue, with the other eigenvalue(s) taking a large negative value. The eigenvector corresponding to the small eigenvalue should represent the axis travelling along the vessel, as the image intensity should be approximately constant in this direction, with the remaining eigenvectors defining the cross-section. This observation has led to these methods still providing useful tools to researchers and providing the basis for many more advanced vessel techniques.

Vascular segmentation is also closely related to the task of vessel extraction. In this case we are interested not necessarily in a pixel-wise labelling of the image, but rather some description of the topology and geometry of the vascular network itself. Tracking methods have been most widely been applied to the task of neuron tracing. In this area they have been driven forward by community projects such as the DIADEM challenge (Gillette et al., 2011) and more recently the BigNeuron project (Peng et al., 2015). Methods in this area tend to follow either a 'tracking' approach, where the vessels are iteratively traced out from some source, or a graph completion approach where an over-complete graph is produced by detecting critical

points, and then pruned to leave the correct vessel network (Turetken et al., 2013, 2016; Almasi et al., 2015, 2016).

2.2.1 Literature

Two thorough reviews of vascular segmentation methods have been produced by Kirbas & Quek (Kirbas and Quek, 2003) and later by Lesage (Lesage et al., 2009). A review on the visualisation of tumour vasculature and its applications has been produced by Missbach-Guentner *et al.* (Missbach-Guentner et al., 2011). A summary of successful methods applied to the VESSEL12 challenge data can also be seen in the write-up by Rudyanto *et al.* (Rudyanto et al., 2014).

The fundamental insight, central to most vessel segmentation algorithms, comes from an analysis of the eigenvalues of the locally computed Hessian matrix. It was observed that tubular structures are defined by certain regimes of curvature on the scalar field defined by an image. These regimes can therefore be identified by an analysis of the local Hessian eigenvalues. The classical eigenvalue methods are those of Frangi (Frangi et al., 1998), Lorenz (Lorenz et al., 1997), Sato (Sato et al., 1997) and Krissian (Krissian et al., 1998). In these papers, the idea of eigenvalue analysis of the Hessian matrix are developed in order to characterise and detect tubular objects in images, namely vessels.

Differential geometry tells us that the eigenvectors of the surface shape operator corresponds to the directions of principal curvature, with the corresponding eigenvalues giving the respective magnitude. This provides a natural coordinate system for a local filter which is independent of the initial reference frame. For a surface of the form

$$S = (x, y, f(x, y)), \tag{2.2}$$

this shape operator takes the form of the Hessian matrix H:

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}. \tag{2.3}$$

Eigen-decomposition of the Hessian matrix provides an elegant alternative to a brute force search for directions of notable response through steerable filters (Freeman and Adelson, 1991), or can augment their use by preselecting salient directions (Sofka and Stewart, 2006). Most methods work on the hypothesis that a tubular region consists of a single, low curvature direction, with two large, negative directions of curvature. The classical formulation of 'vesselness' is given by a combination of exponential responses to certain eigenvalue criteria (Frangi et al., 1998), where $\{\lambda_1, \lambda_2, \lambda_3\}$ represents the set of eigenvalues, ordered by their absolute value (λ_1 possessing the smallest absolute value):

$$V_\sigma(s) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0, \\ \left(1 - \exp\left(-\frac{R_A^2}{2\alpha^2}\right)\right) \exp\left(-\frac{R_B^2}{2\beta^2}\right) \left(1 - \exp\left(-\frac{S^2}{2c^2}\right)\right) & \text{else} \end{cases} \tag{2.4}$$

$$R_A = \frac{|\lambda_2|}{|\lambda_3|}, \tag{2.5}$$

$$R_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}}, \tag{2.6}$$

$$S = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} = ||H||_F. \tag{2.7}$$

This can be interpreted as returning a strong response if:

1. λ_2 and λ_3 are approximately equal. This means the ellipsoid has an approximately circular cross section (tubular vessels).

2. $|\lambda_1|$ is considerably smaller than both $|\lambda_2|$ and $|\lambda_3|$. This enforces the original tubular hypothesis of low curvature in one dimension.

3. The Frobenius norm of the Hessian matrix is large. The vessel is an inherently 2^{nd} order structure so there should be good Hessian strength.

A number of methods incorporate these Hessian based descriptions into different methods, to target them specifically to vessel segmentation. In order to increase intensity regularity along vessel-like structures, an anisotropic diffusion scheme oriented and scaled according to these vesselness considerations was proposed

by Manniesing (Manniesing et al., 2006a). A diffusion tensor was generated for each voxel to encourage image intensity to diffuse along vessels in regions of high vesselness and to diffuse isotropically in non-vessel regions. The diffusion tensor was constructed from the orthogonal basis and eigenvalues generated by the Hessian eigen-decomposition at each voxel. Vessel orientation from Hessian eigen-decomposition has also been used to augment an active contours methods utilizing flux (Gooya et al., 2012). Although considerations on curvature form the basis of Hessian filters, additions have been made such as the incorporation of edge suppression terms (Sofka and Stewart, 2006).

The work of Krissian *et al.* (Krissian et al., 1998) attempted to derive an analytic relationship between the vessel boundary and the scale at which it is detected, but this work relied heavily on model based assumptions on the vessel profile, namely that it will be Gaussian. This can also be related to the work of Pizer *et al.* (Pizer et al., 1994) on 'medialness' filtering and medial representations.

Work by Xiao *et al.* (Xiao et al., 2011) extended the concept of Hessian filtering by considering an analogy to the stress tensor of solid mechanics. By formulating the problem in terms of the tensor invariants, they were able to derive rigorous justifications for the eigenvalue criteria. They made use of a measure for strain energy from solid mechanics as well as a novel 'shape mode' measure. This represented a significant step forwards in one of the key weaknesses in the methods of Frangi *et al.*, namely the enhancement of bifurcations, as they do not obey the basic curvature assumptions.

The summary of methods submitted to the VESSEL12 lung vessel segmentation challenge (Rudyanto et al., 2014) show the state-of-the-art by 2012. The methods submitted by various challengers are dominated by vesselness measures, still largely derived from the works of Frangi (Frangi et al., 1998) and Krissian (Krissian et al., 1998) as well as more modern filters such as the work of Xiao *et al.* previously mentioned (Xiao et al., 2011). However, the winning method by Kiros *et al.* (Kiros et al., 2012) used stacked convolutional filter learning, coupled with a logistic regression, machine learning approach. In this method, convolutional kernels

were learned using a k-means based approach, outlined in the work by Coates *et al.* (Coates and Ng, 2012). Although this challenge occurred slightly before the breakout success of Convolutional Neural Networks in the ImageNet challenge (AlexNet (Krizhevsky et al., 2012)), the principals behind this submission are similar. By learning hierarchical convolutional features it is possible to project images into an extremely discriminative higher dimensional space. From this space, simple linear classification methods, such as logistic regression can be used on a pixel-wise basis to perform segmentation. Although this principal has been applied for a long time using hand-crafted features, results such as those from these two challenges, show the power of learning domain specific hierarchical features.

The vast majority of vessel segmentation research is directed towards the segmentation of the vasculature of the retinal fundus. Although not directly applicable to our task, the STAR and DRIVE datasets provide a way to quantify methods on this application. The current state-of-the-art methods rely heavily on deep convolutional neural networks, such as in the current leading work by Liskowski *et al.* (Liskowski and Krawiec, 2016). Other deep learning approaches to segmentation in this area include work by Fang *et al.* (Fang et al., 2015) and Li *et al.* (Li et al., 2016). Prentašic *et al.* have applied deep learning methods to coherence microscopy images of retinal micro-vasculature (Prentašic et al., 2016). In the work by Liskowski *et al.* , they train a deep CNN to segment retinal vascular networks. Although they achieve state-of-the-art results on this particular data set, the network configuration they use is a patch-based CNN rather than the current state-of-the-art for segmentation which is largely dominated by fully convolutional networks. This introduces an inefficiency in both training and execution. In order to achieve these results they make use of a number of augmentation techniques such as Zero-Phase Component Analysis (ZCA) whitening (Kessy et al., 2017), contrast normalisation, image rotation, image scaling, image mirroring and gamma/saturation modification.

Teikari *et al.* have applied CNNs to the task of segmentation in 3D fluorescence microscopy images of micro-vasculature, similar to ours (Teikari et al., 2016). However, they also consider vasculature that has been contrast enhanced. In this

work they utilise the ZNN framework of Lee *et al.* (Lee et al., 2015). The ZNN network incorporates a hybrid 2D-3D convolutional network that is implemented on the CPU. This reliance of CPU computation is due to the inherent memory consumption for 3D CNNs applied to large image volumes. For their proposed VD2D3D network they suggest a 'recursive' training strategy that involves training the 2D and 3D parts of the network independently before concatenating them.

2.3 Supervoxels

Supervoxels, or superpixels for non-volumetric images, are a family of popular spatial clustering techniques, applied to images. The central concept is to identify localised, contiguous clusters of voxels that are either all part of the same object in the image, or share a degree of homogeneity. By doing this we are able to describe the image as the set of all supervoxels, rather than the set of all voxels. If the supervoxels have been well assigned then this should not result in a loss of specificity, as any assignment to a voxel should be shared with all voxels in the same supervoxel. In this sense they can be thought of as capturing redundancy in an image, allowing us to massively reduce the dimensionality of any problem applied over the whole image. For example, an image containing 10,000,000 voxels might be accurately summarised by 10,000 supervoxels, eliciting a 1000 times reduction in dimensionality for any labelling problem.

A number of authors have proposed methods for such a clustering. Engel *et al.* applied the classical Watersheds algorithm (Engel et al., 2009), Shi and Malik have used eigenvalue considerations in their Normalized Cuts method (Shi and Malik, 2000). The Graph Cuts algorithm of Boykov *et al.* for solving sub-modular optimisation problems (Boykov et al., 2001) has been applied to supervoxels by Felzenszwalb (Felzenszwalb and Huttenlocher, 2004). Comaniciu and Meer proposed a supervoxel method using Mean Shift clustering (Comaniciu and Meer, 2002). However, one of the most popular methods has come from a fairly recent work by Achanta *et al.*, named Simple Linear Iterative Clustering (SLIC) (Achanta et al., 2012). They assign cluster centres using a weighted spatial and feature-space

distance metric. Similar to k-means clustering, SLIC utilises an Expectation-Maximisation (EM) framework, where first, each voxel is assigned to its most likely (in this case closest) cluster center. Next, each cluster center is moved to the 'most likely' location, given its assigned voxels. In the case of SLIC, this means assigning to the centroid of the assigned voxels, minimising a least squared distance. This is iterated upon until convergence and the voxels assigned to a cluster center define a single supervoxel. Cluster centres are usually initialised on a regular grid. k-means clustering of this kind makes the assumption of spherical clusters, however, due to the packing of large numbers of cluster centres, the natural state for SLIC superpixels is a square or hexagonal regular lattice.

2.3.1 Literature

Two of the most popular methods in supervoxel generation are Felzenszwalb's Graph Cuts based method (Felzenszwalb and Huttenlocher, 2004) and Achanta's Simple Linear Iterative Clustering (SLIC) method (Achanta et al., 2012). While Felzenszwalb's method tends to create supervoxels which contain entire objects within the image, SLIC generates uniform and local clusterings which respect image boundaries. SLIC performs a k-means (MacQueen, 1967) clustering where the metric is a combined spatial and intensity distance. As we wish to use supervoxels for local information sharing, the SLIC method is preferable. In fact, it has been suggested that averages over SLIC supervoxels can be considered a form of bilateral filtering (Heinrich et al., 2013). Although most applications of supervoxels have come from the computer vision community in two-dimensional applications (superpixels), supervoxels have also been applied to biological tasks in three-dimensions.

The principle of using supervoxels to segment elongated dendritic structures was demonstrated by Kim *et al.* (Kim et al., 2014), although the decisions on inclusion of individual supervoxels were performed by volunteers through a form of 'gamification', EyeWire[6].

[6]https://eyewire.org

In addition to applying superpixels to image intensities, a number of other authors have considered the computation of superpixels on transformed signal domains. Irving *et al.* have computed superpixels using the principal components of Dynamic Contrast Enhanced MRI (DCE-MRI) uptake kinetics for tumour segmentation (Irving et al., 2014b). Oda *et al.* have used the eigenvalues of Hessian matrix as an additional source of information for superpixels when segmenting lymph nodes in CT (Oda et al., 2017). Holzer *et al.* have utilised the monogenic phase for computing supervoxels for medical images (Holzer and Donner, 2014).

To our knowledge, only one other work has applied supervoxels to the segmentation of vasculature in 3D (Zhu et al., 2013). This is an extension of previous work performed in 2D (Zhu et al., 2011). In an attempt to produce a supervoxel based vessel segmentation method they have considered averaged orientations and relative intensities over supervoxels. In this work, they consider a SLIC supervoxel segmentation and consider relative mean intensity of a supervoxel rI_j defined as:

$$rI_j = \max_{k \in \mathcal{N}_j} (I_j / I_k) \tag{2.8}$$

where $\mathcal{N}_j$ is the set of neighbouring supervoxels to the jth supervoxel and I_j is the mean intensity in the jth supervoxel. This is used as a method of improving contrast between foreground and background structures. A major flaw in this metric is that it assumes that every foreground supervoxel has at least one background voxel as a neighbour. This could be encouraged by correctly tuning the size of the supervoxels to the diameter of the target vessel, but if there are vessels of many different sizes then this will be impossible. The proposed implementation uses SLIC supervoxels, which are known to produce regularly sized supervoxels. If the size of the supervoxels is tuned to the smallest visible vasculature then the relative intensity metric will fail due to interior supervoxels.

In addition to relative intensity, they consider a metric for directionality over a supervoxel. They calculate supervoxel directionality as the vector formed from the sum of all voxel-level directions, calculated from Hessian eigenvectors. They hypothesise that, if a supervoxel contains a vessel, then the directions within it will

be highly correlated and thus, this summed vector will be large. If a supervoxel contains background, then the directions will be randomly distributed due to noise and thus, the size of the summed vector will tend to zero by the Central Limit Theorem. However, this ignores the fact that directions may be correlated by a number of other image features. When smoothing at large values of σ, directions tend to be highly locally correlated, even in areas of noise alone. In addition to this, summing these directional vectors directly ignores the fact that both v and $-v$ are equally valid descriptors of the same direction of curvature.

A large body of literature also includes the combination of supervoxels with graphical models, which will be addressed in the next section.

2.4 Graphical Models

Although feature extraction represents a way to encode some degree of spatial interaction between pixel values, often we wish to make this interaction more explicit. Graphical models allow us to encode a spatial prior over our imaging tasks and have been extremely popular in segmentation, as well as other areas of imaging. We can consider any decision in a labelling task based on just the features extracted as a unary decision, a decision that is made in isolation. Markov Random Fields (MRF) define a joint probabilistic model defined over a graph (most simply the graph of pixels, with edges connecting neighbouring pixels) which combine the unary decisions at each node in the graph with pairwise (and higher order) interactions. Although subtle differences in definition exist between MRFs and Conditional Random Fields (CRFs) often the nomenclature is used interchangeably.

The application of graph-based segmentation images originates with the work of Besag (Besag, 1974), where he observes that the (then recently proven) Hammersley-Clifford theorem gives conditions on the functional form of the conditional probabilities over the graph such that they result in a consistent joint probability over all factorisations. This functional form is the Gibbs distribution and is as follows:

$$P(x) = \frac{1}{Z}\exp(-E(x)) = \frac{1}{Z}\exp(-\sum_{c} E_c(x)) \tag{2.9}$$

Subsequently, the joint probability can be factorised over the cliques c of the graph. Defining the joint probability in terms of the potentials E provides a much more intuitive formulation of the MRF. If we consider only pairwise cliques then, on a graph defined by its set of vertices and edges, $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, this usually takes the form

$$E(x) = \sum_{i \in \mathcal{V}} D(x_i) + \sum_{(i,j) \in \mathcal{E}} V(x_i, x_j). \tag{2.10}$$

From Equation 2.9 it follows that we can estimate the maximum a posteriori (MAP) solution for the joint probability by minimising this potential V instead. This transforms the problem from a probability maximisation to an energy minimisation.

The seminal application comes from the work of Geman and Geman (Geman and Geman, 1984), in the area of image restoration. They consider a suitable Markov model to repair noise corrupted images as well as proposing the popular Simulated Annealing (SA) method to minimise the energy. They also provide a cooling schedule for SA that guarantees its convergence to a global minimum. The SA method requires that we sample from highly multidimensional distributions, and the authors also provide an algorithm for doing this, the Gibbs Sampler. Although this is an extremely influential piece of work, simulated annealing can be extremely slow to converge for larger problems. In addition to this, ergodicity, and thus a global optimum, can not be guaranteed in finite time.

The development of algorithms to solve the min-cut/max-flow problem on graphs in polynomial time has expanded the popularity of graphical segmentation methods, the most influential being the Graph Cuts algorithm of Boykov *et al* (Boykov and Jolly, 2001). The min-cut/max-flow problem formulates this energy minimisation over the label space as a flow problem through a graph with addition 'source' and 'sink' nodes added for the labels. Letting a graph cut $\mathcal{C}$ be a set of edges such that the induced graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} - \mathcal{C} \rangle$ has no connection between the 'source' and 'sink' nodes. The min-cut/max-flow theorem shows that the maximal flow through the graph is equal to the capacity of the minimal cut. The work of Boykov shows that the

minimal cut on an appropriately defined graph is equivalent to the minimal labelling of our graph such that the energy V of Equation 2.10 is minimised. In addition to this they provide two algorithms, α-expansions and α-β swaps, to approximate this minimal cut over the label space in polynomial time. They show that a solution can be found within a constant factor of the global minimum. The stated requirement on the pairwise term W for this to hold is that it must be sub-modular.

The special case of image segmentation is a two-label formulation of this problem and further work from Boykov and Funka-Lea has shown that the global optimum can be found for a two-label problem with a sufficiently regular pairwise term (Boykov and Funka-Lea, 2006). For the two-label problem it suffices that

$$V(\alpha, \beta) = 0 \text{ iff } \alpha = \beta \tag{2.11}$$

and

$$V(\alpha, \beta) + V(\beta, \alpha) \geq 0 \tag{2.12}$$

Much work has gone into determining exactly what can be encoded within these unary and pairwise terms. The most basic pair-wise term is that of the Potts model, employed by Geman and Geman (Geman and Geman, 1984),

$$V_{pq}(f_p, f_q) = \begin{cases} K & \text{if } f_p \neq f_q \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

This pairwise term attempts to encode the fact that regions of images tend to be homogeneous, that is to say, pixels more often than not represent the same thing as their neighbour. This serves to eliminate outliers but tends to create regions with smooth, rounded boundaries and small, thin objects such as vessels can be smoothed away.

Attempts to encode anisotropy into these terms have been made using surface flux in a pure Graph Cuts sense (Kolmogorov and Zabih, 2004; Kolmogorov and Boykov, 2005). Here they provide a graph construction to optimise flux using graph-cuts. The construction in this paper is applied to a gradient vector field as well as a vector field derived from a prior model. However, the construction could

be applied to optimise a flux derived from any vector field. This flux description is encoded in the unary potentials of the graph nodes.

2.4.1 Literature

In work by Kaynig *et al.* an anisotropic MRF combined with a support vector machine (SVM) classifier is used to segment thin, elongated, membrane structures in SSTEM images (Kaynig et al., 2010, 2015). The SVM is trained on manual annotations and learns high dimensional feature representations for the unary terms. They suggest an alternative formulation to flux maximisation, by encoding the concept of flux into the pairwise interaction term. They suggest that minimising flux energy in the unary term, in the style of Kolmogorov (Kolmogorov and Boykov, 2005) is prone to false positives due to noise. Anisotropy has also been encoded into the pairwise term by considering a 'Coherence tensor' (Grau et al., 2006) derived from the eigenvalues of the structure tensor (Weickert, 1999). In this work by Grau *et al.* they utilise an anisotropic MRF for the segmentation of the optic nerve head in glaucoma. They find that the introduction of an anisotropic regularisation scheme improves robustness to a number of parameters, including MRF strength. Anisotropic MRFs have also been used by Bériault *et al.* (Bériault et al., 2014) for the improved segmentation of vasculature in susceptibility-weighted imaging (SWI) venography images.

There is a huge amount of synergy between graphical segmentation schemes and supervoxel methods. The supervoxels provide a reduced graphical representation of the image, making even the largest image volumes potentially amenable to a Graph Cuts optimisation or similar. The supervoxels also allow a much richer set of features to be produced, which may be employed in terms of unary or pairwise potentials to be optimised. Graphical methods, such as MRFs, provide a strong framework in which relationships between supervoxels can be encoded.

Work by Fulkerson *et al.* (Fulkerson et al., 2009) combined superpixels and MRF models for object detection. In this work, an SVM model applied to image features accumulated over each superpixel provides the 'unary' term for the MRF.

The pairwise interactions are formed by a combination of feature comparisons as well as superpixel shape comparisons, such as the size of the shared border. Pixel-level features are extracted using SIFT, and are then quantised using a k-means approach. These quantised features are then aggregated over each superpixel to form an L_1 normalised feature vector for each superpixel. In order to regularise results, each superpixel is assigned a feature vector that is the average of all neighbouring superpixels' feature vectors as well as its own.

This concept was taken further by Lucchi *et al.* (Lucchi et al., 2012), who applied supervoxels to the segmentation of mitochondria from electron microscopy image volumes. They propose an MRF graph-partitioning scheme, defined over the graph of supervoxels, that uses both textural and shape-based features. To describe supervoxel shape, they propose a 'Ray-descriptor', which looks at the distance to the boundary in all directions from the centre of the supervoxel. In order to make this feature comparable between supervoxels, a frame of reference is established according to the first two PCA modes of the Ray-descriptor 'terminal points'. A concatenation of supervoxel Ray-descriptors with the descriptors of their neighbours is used. In addition to the shape features, textural features are encoded as a histogram of intensities within the supervoxel, concatenated with a histogram of neighbour intensities. These features are used to train an SVM classifier. For their graph partitioning scheme they consider a three-labelled problem: Background, Boundary and Mitochondria. Using a manually annotated training set they are able to train probabilities for each label as well as each pair of labels. These can be re-purposed as unary and pairwise potentials in the MRF scheme.

Superpixels have also been used in higher order graphical models, in work by Kohli *et al.* (Kohli et al., 2009). In this work, superpixels were used to provide a soft labelling constraint on pixels in an image. This is distinct from the majority of superpixel/MRF methods, which treat each superpixel as a distinct entity and attempt to assign a label to each.

2.5 Machine Learning

Machine learning is described broadly as the process by which some model M may improve its ability to perform task T through the exposure to some set of training examples D. This broad definition applies to many concepts from statistics, optimisation and model fitting. However, while these fields are related, a distinction may be drawn from the fact that, in machine learning, we are largely interested in the model's ability to be predictive rather than the values that the fitting parameters take. In modern parlance, machine learning largely refers to a common set of techniques, where flexible models, whose parameters hold no intended meaning, are fit to data sets. The two most common tasks T for algorithms of this kind are regression where we attempt to predict some value from $\mathbb{R}^n$ or classification, where we attempt to predict the membership of some category from the set $C = \{c_1 \dots c_n\}$.

In this thesis we will employ the use of two important algorithms in the world of machine learning, decision forests and neural networks.

2.5.1 Decision Forests

Decision forests natively exist in the world of categorical classification, however, they are an extremely flexible formalism that have been tasked to the entire range of problems within the world of machine learning. In order to understand decision forests we should break them down into a hierarchy of their component parts. The 'forest' consists of a number of 'trees' and the 'trees' are in turn made up of a branching structure of 'splitting nodes' and 'leaves'. The splitting nodes, in full generality, partition the set of inbound data points into two sets, a left set and a right set:

$$\{D_l, D_r\} = \text{Split}(D), \tag{2.14}$$

$$D_l \cap D_r = \emptyset, \tag{2.15}$$

$$D_l \cup D_r = D. \tag{2.16}$$

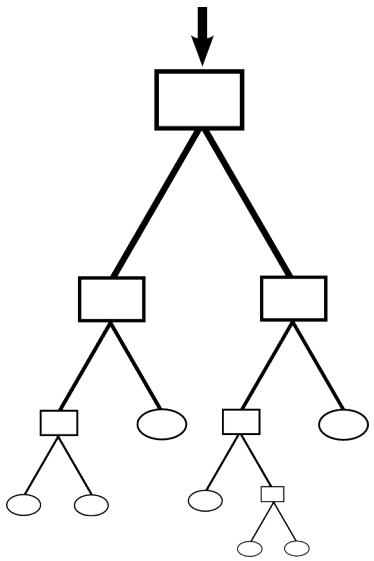


Figure 2.7: Recursively branching decision tree. Rectangles denote splitting nodes and ovals denote leaf nodes.

In practice, the splitting nodes often take the form of a parametrised threshold t applied to a single data dimension.

$$D_l = \{x : \ x < t \quad \forall \quad x \in D\} \tag{2.17}$$

$$D_r = \{x : \ x \geq t \quad \forall \quad x \in D\} \tag{2.18}$$

where t is learned for each node. With this approach we can think of each tree as defining a partitioning of the input space by a number of axis aligned planes, each partition is then assigned some output value. The tree structure continues to branch until a leaf node is reached, which dictates the 'output' of each tree, given the input data point. In a categorical sense the leaf node may represent a single category, or more generally, a distribution over categories. In the case of regression the leaf node could produce a value from $\mathbb{R}^n$ or a function that maps the inbound data points to $\mathbb{R}^n$.

The 'tree' therefore consists of the collection of these connected split nodes and leaf nodes. A data point that is passed to this tree will take a path through the tree, depending on the various split node decisions that it passes. An example of the hierarchical structure of split nodes and leaf nodes is shown in Figure 2.7.

In order to produce a 'forest' from these trees we simply consider a collection of independent trees. In order to provide a single prediction from the forest we combine the 'tree' outputs in some sense. Either by averaging regression values, or taking some average distribution over categories or by taking the modal categorical prediction.

We will now briefly outline the training procedure for these decision forests and finally discuss some of their properties. We refer to the process of fitting this model to our data as 'growing', which is fitting given the floral nomenclature in use.

When training a tree, we will begin by fitting a splitting node to our initial data points. Then for each split (left and right) we will check if a stopping condition has been met. If not then we will fit another splitting node to this split and so on. If a stopping condition has been met then we fit a leaf node instead. In order to fit a splitting node we attempt to partition the input subspace such that some measure of information on the resultant splits is maximised. In essence, we wish to produce homogeneous subsets of the training data in each partition, such that we may feasibly assign a single label to all data points that lie within this partition.

The two most commonly used measures of information gain are the Gini impurity and the Shannon entropy. In both cases, the gain in information is quantified by

$$I_{gain} = H(D) - \frac{1}{2}(H(D_l) + H(D_r)), \tag{2.19}$$

where

$$H_{gini} = \sum_{i=1}^{|C|} p_i(1 - p_i) \tag{2.20}$$

and

$$H_{ent} = -\sum p_i \log(p_i), \tag{2.21}$$

where p_i is the fraction of the subset of data that are labelled as category i. In both cases we see an information gain if we are able to divide the data points into two subsets who's average homogeneity is greater than before.

In order to evaluate the decision tree we simply follow the path directed by the splitting node until we arrive at a leaf node, the output is then decided by

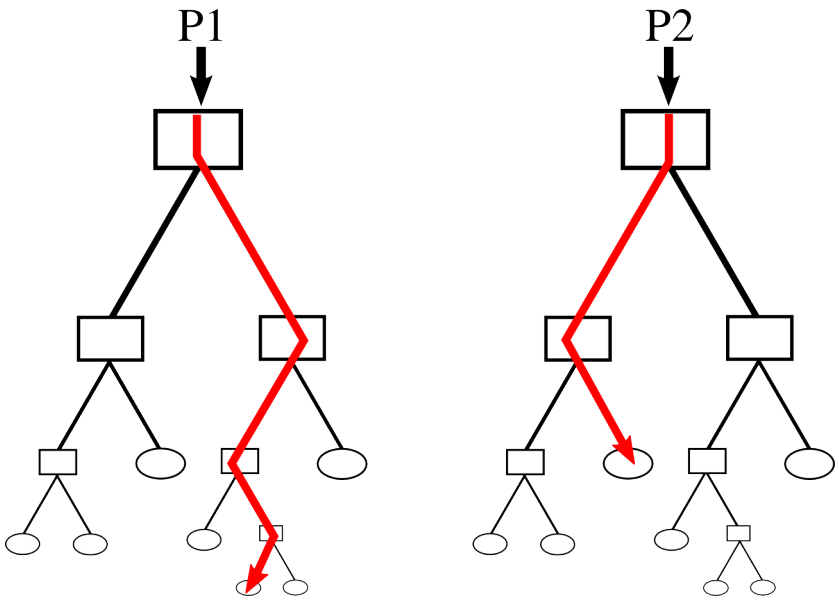


Figure 2.8: Different data points result in differing routes through the tree, ultimately arriving at different leaf nodes.

this leaf node. An example of the different paths taken by two different input data points is shown in Figure 2.8.

The most popular incarnation of the decision forests formalism is the Random Forests (RF) approach of Breiman (Breiman, 2001) and Ho (Ho, 1998) who both observed the power of using Bagging (Bootstrap-Aggregation) to encourage the decorrelation of errors between individual trees of a forest. In order to achieve this, each tree in the forest is trained on some sub-sample of the training set. It is possible to show that an ensemble of decorrelated predictors has a lower variance of error than each single predictor alone, making highly resistant to over-fitting. The particular kind of subset used is a sample from the Canonical Distribution of the data. This is achieved by uniform sampling with replacement. In order to further reduce the correlation between the trees, they also suggest making only a subset of the features available for splitting at each splitting node. Although the No Free-Lunch Theorem tells us that no single classifier can out-perform any other classifier when averaged over all possible data sets (Wolpert, 1996), in practise some will clearly be more effective than others on real world data. Fernandez-Delgado *et al.* (Fernández-Delgado et al., 2014) compared the effectiveness of 179 different

classifiers from 17 families of classifier, tested over 121 datasets. In this study they found the Random Forest family of classifiers to be the most effective, although not significantly better than the SVM family.

2.5.2 Neural Networks

Neural networks, also known as feed-forward neural networks or artificial neural networks (ANN), are another important class of machine learning algorithms and are loosely inspired by the behaviours of the biological axons found in the brain. The principal behind neural networks is that complex functions can be approximated by the composition of many simpler functions.

An ANN is typically described by a number of 'nodes' and a Directed Acyclic Graph (DAG) connecting these nodes. A DAG is a directed graph which contains no loops. The significance of this is that every possible path through the network is finite and arrives at some terminal node. This leads to the terminology of a feed-forward network as it is possible to propagate some computation forward from the 'input' nodes until we reach our 'output' nodes, where each computation performed at a node is dependent only on the results of the computations performed at the nodes which connect to it. We consider all nodes with no inbound edges to be 'input' nodes and it is here that we must feed values into the network. We consider all nodes with no outbound edges to be 'output' nodes and it is here that the ultimate output of our forward pass will be produced.

The network is parametrised by a set of weights, one for each edge, and a set of biases, one for each node. A common computation that may be performed at the ith node is given by

$$f_i(x; w) = \sigma\left(\sum x_j w_{ij} A_{ij} - b_i\right), \tag{2.22}$$

where A is the adjacency matrix for the DAG. This means that for each node we take an inner product between the vector of values at all inbound nodes with the vector of weights for all inbound edges, adjust this by some bias term and then modulate this value by some non-linear function σ. From this formulation we see

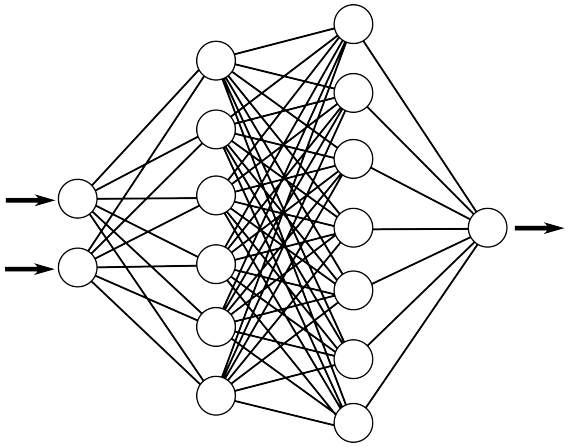


Figure 2.9: A simple neural network model with two hidden layers.

that it is possible to also consider the bias value to be a weight attached to some constant input value of 1. For the remainder of this section we will include bias in our discussion of weights. Although the values x_j in this equation themselves rely on the value of the computation f_j, as our graph is acyclic we are guaranteed to eventually arrive at input nodes as we traverse the graph backwards. Since we provide the values to all input nodes then we are able to begin computation there.

Neural networks are typically arranged in layers, with all nodes on a given layer connected to all nodes on the subsequent layer but no connections within a layer. The preferred nomenclature for a layer of this kind is a 'fully connected' layer, often denoted *fc*. Layers that contain neither the input nodes nor the output nodes are known as 'hidden layers' as the representations formed in these layers are never directly observed, although they are crucial to the mapping between the input and output layers. A network architecture with two input nodes, one output node and two hidden layers can be seen in Figure 2.9.

The model is therefore described by the set of weights w and the adjacency matrix A which describes what we will refer to as the 'network architecture'. When fitting a model such as this we keep the architecture constant while optimising w. We define the combined action of our network on some input data x as $f(x; w)$. In order to fit this model to some task T given data D we must define some loss function $\mathcal{L}$ that describes how well we are doing. If we consider D to be composed

of a number of input and target pairs (x, t) then

$$\mathcal{L} = g(f(x; w), t). \tag{2.23}$$

Therefore $\mathcal{L}$ describes some discrepancy between the network output for the input value x and the correct output t. The aim of fitting a neural network is to find the set of weights $\bar{w}$ that optimises this loss function $\mathcal{L}$:

$$\bar{w} = \arg\min_{w} \frac{1}{|D|} \sum_{(x,t)\in D} \mathcal{L}(f(x; w), t) \tag{2.24}$$

As we constructed $f(x; w)$ via the composition of a number of differentiable functions, assuming g itself is differentiable we are able to differentiate $\mathcal{L}$ and determine $\frac{\partial \mathcal{L}}{\partial w_i}$ through application of the chain rule. We can then optimise our weights via methods of gradient descent. This technique is known as back-propagation and we will discuss it in greater detail later on.

Loss Functions

In the case of a regression task, we will most often optimise a Mean Squared Error (MSE) loss. In this case we view the non-final layers as building a representation of the input data. The final layer can then be viewed as a linear regression. We can show that optimising a MSE loss, under the assumption that our regression targets $\mathbf{t}$, follow a linear model in this transformed space, with errors distributed as Gaussian random variables with some variance β, is the Maximum Likelihood Estimator (MLE). Letting

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}) = \prod_{i}^{N} p(t_i|x_i, \mathbf{w}) \tag{2.25}$$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}) = \prod_{i}^{N} \frac{1}{\sqrt{2\pi\beta^2}} e^{\frac{(f(x_i,\mathbf{w})-t_i)^2}{2\beta^2}} \tag{2.26}$$

then taking logarithms,

$$\ln(p) = \frac{1}{2\beta^2} \sum_{i}^{N} \left(f(x_i, \mathbf{w}) - t_i\right)^2 + \frac{N}{2} \ln(\beta^2) + \frac{N}{2} \ln(2\pi). \tag{2.27}$$

Since we are optimising $\mathbf{w}$ it therefore suffices to consider the MSE term only. Although in the case of linear regression this leads to a convex optimisation problem

that can be solved globally, the functional form of f will invariably lead to this problem being non-convex. This means that we are not guaranteed to find the global minima and in practice will end up at some local minima.

In the case of a classification task we are no longer attempting to predict values in $\mathbb{R}^n$ as we are attempting to predict a categorical distribution over the categories. This means that any output should exist on the n-dimensional simplex.

$$\{x \in \mathbb{R}^n : x_0 + \cdots + x_{n-1} = 1, x_i \geq 0, i = 0, \ldots, n-1\}. \tag{2.28}$$

Since our variables only exist on this subset of $\mathbb{R}^n$ a different notion of distance is required.

The Kullback-Leibler Divergence (KLD) is a measure between probability distributions and takes the form

$$D_{KL}(P||Q) = \sum_i \left(P_i \log \left(\frac{P_i}{Q_i} \right) \right). \tag{2.29}$$

Alternatively we can rewrite this as

$$D_{KL}(P||Q) = \mathbb{E}_{x \sim P} \left[\log(P(x)) - \log(Q(x)) \right], \tag{2.30}$$

which shows us that this represents the expectation of the difference in log probabilities over the input distribution.

It should be noted that the KLD is not symmetric

$$D_{KL}(P||Q) \neq D_{KL}(Q||P). \tag{2.31}$$

Considering the strict convexity of the function $-\log x$ and Jensen's inequality

$$f\left(\sum \lambda_i x_i\right) \leq \sum \lambda_i f(x_i), \tag{2.32}$$

where $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ describes a probability distribution. Since f is strictly convex then we have equality if and only if $f(x_i) = 0$ for all x_i. Applying this to the KLD and taking $f(x_i) = -\ln(P(x_i)/Q(x_i))$ we get

$$-\sum \left(P_i \ln \left(\frac{P_i}{Q_i} \right) \right) \leq \ln\left(\sum Q_i\right) = 0. \tag{2.33}$$

This implies equality if and only if $P_i = Q_i$ for all i. If we consider our two distributions P and Q as t and $f(x; \mathbf{w})$ respectively, then writing KLD as

$$D_{KL}(t||f(x;\mathbf{w})) = -\sum((t_i \ln(f(x_i;\mathbf{w})) - t_i \ln(t_i)) . \tag{2.34}$$

As t is a fixed distribution relative to the weights w, it remains to optimise the first term only. This is known as the categorical cross entropy:

$$H(f(x;\mathbf{w})) = -\sum t_i \ln(f(x_i;\mathbf{w})). \tag{2.35}$$

Still, it remains to be seen that this is a sensible quantity to optimise. Again we consider the likelihood function for predictions on our data set ($\mathbf{x}$, $\mathbf{t}$), but this time we assume that elements from $\mathbf{t}$ follow a categorical distribution. For simplicity we will first assume that $y_i = f(x_i; \mathbf{w})$ follows a Bernoulli distribution and $t_i \in \{0, 1\} \forall\, i$, therefore,

$$p(\mathbf{t};\mathbf{w}) = \prod_i y_i^{t_i}(1 - y_i)^{1-t_i}. \tag{2.36}$$

Taking logarithms again we see that

$$-\ln(p) = -\sum_i t_i \ln y_i + (1 - t_i) \ln(1 - y_i), \tag{2.37}$$

which is simply the cross-entropy loss we reached in Equation 2.35, expanded for two categories. When optimising a binary cross-entropy loss function it is common to pass the output of the network through a sigmoidal activation, constraining the output to the range $(0, 1)$. Consider a network output of the form $y_i = \sigma(f(x_i; \mathbf{w})) = \sigma_i$. The binary cross-entropy loss function then takes the form

$$\mathcal{L}_{bce} = -\sum_i t_i \log(\sigma_i) + (1 - t_i) \log(1 - \sigma_i) . \tag{2.38}$$

Considering $\frac{\partial \mathcal{L}}{\partial f}$ we see that.

$$\frac{\partial \mathcal{L}}{\partial f} = -\sum_i \frac{\sigma_i(1 - \sigma_i)t_i}{\sigma_i} - \frac{(1 - t_i)\sigma_i(1 - \sigma_i)}{1 - \sigma_i} \tag{2.39}$$

$$= \sum_i \sigma_i - t_i \tag{2.40}$$

Therefore we can compute the gradient of the loss function with respect to the input of the sigmoid function by a simple comparison between the network output σ_i and the target t_i.

We can easily extend this into the multi-class space by considering a categorical distribution over K classes, such that $t_i = \{t_{i1}, \ldots, t_{iK}\}$. Now the likelihood becomes

$$p(\mathbf{t}; \mathbf{w}) = \prod_i \prod_j y_{ij}^{t_{ij}}, \tag{2.41}$$

giving a log likelihood

$$-\ln(p) = -\sum_i \sum_j t_{ij} \ln y_{ij}. \tag{2.42}$$

In the case of a multi-class cross-entropy loss function a similar reduction to Equation 2.40 can be shown for the softmax activation function

$$\text{softmax}(\mathbf{x}) = \frac{\exp(\mathbf{x})}{\sum_i \exp(x_i)}. \tag{2.43}$$

2.5.3 Deep Learning

Deep learning describes the paradigm by which neural networks, endowed with a large number of hidden layers, are able to model extremely complex and varied phenomena. Although neural networks have existed in many different forms for a number of decades, it is through the set of techniques that fall under the deep learning umbrella that they have become the state-of-the-art across the entire spectrum of machine learning applications.

Although these models often contain upwards of millions of parameters, they display startlingly good generalisation and resistance to over-fitting, considering the scale of over-specification for the models. In this section we will outline a number of the key methodologies that allow Deep Learning methods to succeed as well as outlining the core architectural designs and training paradigms that will be employed in this thesis.

Key Methodologies

Back-Propagation As briefly outlined earlier, we wish to optimise some loss function $\mathcal{L}$ with respect to the weight parameters w_i given some training set Φ consisting of N input and output pairs (x_i, t_i) for $i \in \{1, \ldots, N\}$.

$$\mathcal{L}(\mathbf{x}, \mathbf{t}) = \sum_{i}^{N} \ell(f(x_i; \mathbf{w}), t_i), \tag{2.44}$$

where $f(x_i, \mathbf{w})$ represents the full action of our network f on input value x_i with respect to weights $\mathbf{w}$. In order to find the optimal weights we wish to minimise $\mathcal{L}$ with respect to $\mathbf{w}$.

$$\nabla_{\mathbf{w}} \mathcal{L} = \sum_{i}^{N} \nabla_{\mathbf{w}} \ell(f(x_i; \mathbf{w}), t_i) \tag{2.45}$$

$$\nabla_{\mathbf{w}} \ell = \left(\frac{\partial f}{\partial \mathbf{w}} \right)^{T} \nabla_f \ell \tag{2.46}$$

We therefore must find the Jacobian matrix of our combined network function f. Considering our formulation of a neural network in section 2.5.2 as a DAG, we can break down this function into a series of compositions of simpler functions. Each node in this graph represents a multivariate function with n input variables and m output variables. The chain rule allows us to push derivatives 'backwards' through a chain of composed functions. Since each weight modulates the interaction between nodes of sequential layers we can therefore compute errors with respect to an individual weight in terms of the errors at the layer that succeeds it. Since we can compute the error explicitly at the final layer, we are able to iteratively compute all weights by passing the computation backwards using the chain rule. This is the essence of back-propagation, and by the same logic that our forward-pass of computations will eventually terminate at the output nodes, our backward-pass of derivatives will eventually terminate at the input nodes, meaning that we are able to fully evaluate the Jacobian matrix for the network, with respect to our weight parameters.

Stochastic Gradient Descent Once we have constructed a gradient with respect to all weights in our network it remains to attempt to descend this gradient. Since any evaluation of the gradient via back-propagation is conditioned on a given input-target pair (x, t), we construct our loss by taking an average over all available training examples.

$$\nabla \mathcal{L} = \frac{1}{|D|} \sum_{(x,t) \in D} \nabla \ell(f(x; w), y) \tag{2.47}$$

There is a class of optimisation methods that are designed for optimising losses formulated as a sum, known as Stochastic Gradient Descent (SGD). For each iteration of the algorithm we approximate this full gradient by considering only a subset of the terms of the sum, known as a mini-batch. The basic algorithmic structure of SGD is shown in Algorithm 1.

Algorithm 1 Single epoch of the SGD algorithm

Learning rate = α, batch size = s, batches per epoch = n, weights = $\mathbf{w}$
while $i < n$ **do**
  Sample s input-output pairs (x, y) from training set D.
  $g \leftarrow \frac{1}{s} \sum \nabla \ell(f(x_i; \mathbf{w}), y_i)$
  $\mathbf{w} \leftarrow \mathbf{w} - \alpha g$
  $i \leftarrow i + 1$
end while

Although this algorithm ensures that we are always moving locally in the direction of steepest descent, it may be slow, especially in the presence of consistent but shallow gradients. In order to aid this many methods of gradient descent incorporate a concept of 'momentum'. A naive implementation of momentum in SGD is shown in Algorithm 2. Each update step takes a weighted average of the current gradient evaluation and the previous update step, governed by momentum parameter β. This results in an exponential decay in contributions of historical gradient evaluations to each update step. This means that historically co-linear components of the gradient can accumulate over time while the other aspects of the gradient might be taking largely differing directions.

Algorithm 2 Single epoch of the SGD+Momentum algorithm

Learning rate = α, batch size = s, batches per epoch = n, weights = $\mathbf{w}$, momentum = β
while $i < n$ **do**
  Sample s input-output pairs (x, y) from training set D.
  $g_i \leftarrow \frac{1-\beta}{s} \sum \nabla \ell(f(x_j; \mathbf{w}), y_j) + \beta g_{i-1}$
  $\mathbf{w} \leftarrow \mathbf{w} - \alpha g_i$
  $i \leftarrow i + 1$
end while

The gradient descent algorithm that we will use for all experiments in this thesis is the popular Adam optimiser (Kingma and Welling, 2013). The Adam optimiser incorporates a number of beneficial properties of a stochastic gradient descent algorithm, including momentum, gradient rescaling and maintains an adaptive learning rate for each parameter independently. It can be considered a combination of the previous AdaGrad and RMSProp methods and has seen wide adoption within the deep learning community. An algorithmic overview for the Adam optimiser is shown in Algorithm 3.

Algorithm 3 Single epoch of the Adam algorithm (Kingma and Ba, 2015)

Learning rate = α, batch size = s, batches per epoch = n, weights = $\mathbf{w}$, 1st moment decay parameter = β_1, 2nd moment decay parameter = β_2, 1st moment vector = m_t, 2nd moment vector = v_t, $\epsilon = 0.0001$.
while $i < n$ **do**
  $t \leftarrow t + 1$
  Sample s input-output pairs (x, y) from training set D.
  $g_t \leftarrow \frac{1}{s} \sum \nabla \ell(f(x_j; \mathbf{w}), y_j)$
  $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (biased estimate of 1st moment)
  $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (biased estimate of 2nd raw moment)
  $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (bias correction for 1st moment)
  $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (bias correction for 2nd moment)
  $\mathbf{w} \leftarrow \mathbf{w} - \epsilon \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$
  $i \leftarrow i + 1$
end while

Dropout As mentioned earlier, in the discussion on decision forests, there are a number of desirable properties in running an ensemble of independent instances of an algorithm. Although training an ensemble of large neural networks might be

extremely costly, Dropout attempts to imitate the effects of this (Srivastava et al., 2014). Dropout works by randomly 'switching off' a constantly changing subset of nodes during training. At the start of each mini-batch, each node in the network is turned off with probability p. Conceptually this can be considered to be sampling from the exponentially large set of 'sub-networks' within our full network for each mini-batch. At execution we simulate running these networks in ensemble by scaling up each weight by $1/p$. Although a more practical implementation of this is to scale all weights down by a factor of p during training, so that we do not have to remember to up-scale at execution time. It also prevents nodes from excessive 'co-adaptation' as nodes can not rely on specific neighbours being active. A common choice of value for p is 0.5, as this should produce the greatest number of abstracted sub-networks. We can observe this by considering the number of nodes in each sub-network. If our network contains n nodes, and each sub-network contains approximately k nodes, then the number of possible sub-networks is approximately $\binom{n}{k}$, the binomial coefficient. This is maximised when $k = \frac{n}{2}$. In order to get $k = \frac{n}{2}$ we should turn off half of the nodes at each iteration, therefore $p = 0.5$ appears optimal.

Batch Normalization Another issue that arises from training large networks is the fact that modifications to a single layer in the network will impact the effectiveness of subsequent layers in some sense. Although a network as a whole learns a mapping from the input statistics to the output statistics of the training set, consider an internal layer in isolation. An internal layer of the network learns a mapping from the output statistics of the previous layer. Therefore, during training, the layer is not only attempting to learn some useful representation but it is also attempting to compensate for changes occurring in previous layers. Ioffe *et al.* refer to this as *internal covariate drift* (Ioffe and Szegedy, 2015) and proposed Batch Normalization as a way to address it. Batch Normalization applies the concept of data whitening, which has shown to improve convergence in neural networks when applied to the inputs (LeCun et al., 1998), to each layer of a neural network. In this way layers can focus on learning useful mappings rather than 'co-adapting'

with neighbouring layers. This has been shown to improve stability for higher learning rates, allowing the training of larger networks in shorter times. It has also been suggested that Batch Normalization imparts a degree of regularisation to training and may help guard against over-fitting.

2.5.4 Convolutional Neural Networks

The majority of discussion so far has been on the traditional feed-forward ANN. However, there are a number of reasons why these are not well suited for working with images. In this section we will discuss the Convolutional Neural Network, an algorithm that will be used heavily in this thesis and is the current work-horse of machine learning for computer vision. From now on we will draw a distinction between ANNs and CNNs, the former referring to the networks we have described earlier and the latter referring to the convolutional variety.

The main issue with applying ANNs to image domains is the issue of dimensionality. Consider an image as an input that contains 512×512 pixels. A single fully connected layer of equivalent size would require $512 \times 512 \times 512 \times 512 = O(10,000,000,000)$ trainable parameters. This is clearly computationally infeasible. In addition to the computational limitations this is clearly an inefficient use of model parameters. This would mean that shifting an image by a single pixel would cause an entirely different sequence of activations within the network. Clearly any network capable of efficiently analysing images should be largely invariant to small translations such as these. CNNs address this by learning convolution kernels. Rather than learning the relationships between specific pixels, kernel convolutions allow us to learn local dependencies between pixel values. Repeated application of kernel convolutions can model longer range interactions, however, the 'field-of-view' is only extended linearly with the number of composed convolution operators. In order to efficiently learn representations that extend over larger spatial extents, CNNs typically perform convolutions at a number of scales. In practise this usually means a number of convolutions performed at a single scale followed by a down-sampling operation, repeated as necessary.

Although the extraction of features at multiple scales, using convolutional kernels, has been at the heart of computer vision for a long time, the original conception of the Convolutional Neural Network can be traced back to the Neocognitron of Fukushima (Fukushima, 1980, 1988). The LeNet architecture of LeCun *et al.* was the first to use back-propagation to efficiently train such an architecture in a supervised manner (LeCun et al., 1998). This framework of convolutional neural networks, trained by back-propagation, came to prominence with the success of the AlexNet architecture (Krizhevsky et al., 2012). This success was quickly followed by significantly deeper models, namely the VGG architectures of Simonyan and Zisserman (Simonyan and Zisserman, 2014). The GoogLeNet expanded on this deep architecture by modifying the standard protocol of stacked convolutional layers with their so-called Inception units (Szegedy et al., 2015), that combine parallel streams of convolutions of different kinds before recombining. They claim that such a unit is able to achieve similar representational power with fewer parameters, when compared to stacked convolutions alone. Most recently, the ResNet architecture of He *et al.* (He et al., 2016) introduced the concept of Residual Learning to deep learning. This concept suggests that learning a residual function (the deviation from the identity function) rather than the entire function, allows significantly larger architectures to be trained. Current state-of-the-art algorithms such as the Inception ResNet (Szegedy et al., 2016) architecture combine both of these innovations.

Other techniques for applying convolutions at multiple scales exist however. For example the DeepLab architecture of Chen *et al.* (Chen et al., 2016c) employs 'atrous convolutions' where the convolution kernels themselves are dilated, allowing longer range considerations without sacrificing the resolution of the original image. The standard CNN structure has also been recently extended by the DenseNet architecture of Huang *et al.* (Huang et al., 2016). Here they connect convolutional layers to all subsequent convolutional layers rather than simply the following layer.

Traditionally convolutional networks have been applied to image classification tasks, which map an image to a categorical distribution showing what the image depicted. However, these networks can also be put to the task of semantic

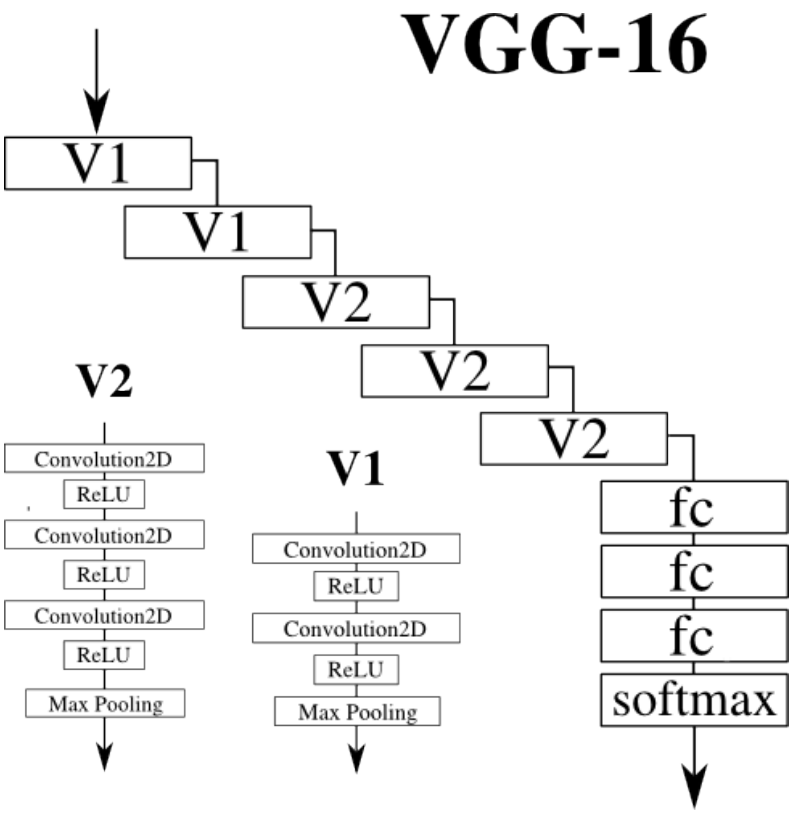


Figure 2.10: Network architecture schematic for the VGG-16 network (Simonyan and Zisserman, 2014).

segmentation. This can be considered to be learning a mapping from an image to a categorical distribution for each pixel, rather than for each image. Initially, image classification networks were applied to semantic segmentation by applying them in a 'sliding window' approach, where small subsections of the image were passed through a network in order to 'classify' the central pixel. However, this is extremely inefficient and imposes an independence between the classification of neighbouring pixels, which is rarely true in practice. A network schematic for the popular VGG-16 CNN (Simonyan and Zisserman, 2014) is shown in Figure 2.10.

Fully Convolutional Networks

Fully convolutional networks (FCNs), popularised by Long *et al.* (Long et al., 2015) in 2015, address the problem of using CNNs to map from one image directly to another. The proposed FCN in this work performs classification at a course pixel-wise level, immediately prior to where the VGG net flattens the images into

fully connected layers, using 1×1 convolutions to flatten the feature maps. This course semantic segmentation is then up-sampled to the original image resolution by learned up-sampling via deconvolution or fractionally strided convolution. This can be considered to be an interpolation with an interpolation kernel that can be learned via back-propagation. The authors crucially observed that simply up-sampling from this course representation alone would prohibit fine detail from occurring in the final segmentation. They therefore proposed to incorporate information from earlier in the network via the use of skip connections. This idea was taken forward by the seminal work of Ronneberger *et al.* (Ronneberger et al., 2015) with their U-Net architecture, which has become ubiquitous in the world of semantic segmentation and medical imaging more generally. The U-Net architecture is best understood as an auto-encoder with skip connections. A schematic of the architecture is shown in Figure 2.11, where we see that the network consists of a number of convolutional and down-sampling 'encoding' layers, followed by a sequence of deconvolutional/up-sampling, 'decoding' layers. Immediately after each up-sampling operation, the feature maps are concatenated with the feature maps of equivalent size from earlier in the network. The crucial difference between this approach and the FCN network is that the feature maps are not collapsed until the end of the network in this case, with classification occurring in the very final layer. An interpretation of this is that multi-scale information is being used to form the decision on classification, rather than just as an aid to reconstruction. The SegNet architecture of Badrinarayanan *et al.* (Badrinarayanan et al., 2016) similarly applies a U-shaped architecture to this problem. This method differs from the work of Ronneberger *et al.* in the fact that the down-sampling and up-sampling are performed in a consistent manner across the skip connections, supposedly reducing the need for learnt up-sampling. Additionally, the DenseNet of Huang *et al.* (Huang et al., 2016) has been extended to a fully convolutional variant by Jegou *et al.* (Jegou et al., 2017), utilising dense blocks of convolutions rather than the standard sequential convolutions.

In training fully convolutional networks such as this we employ a pixel-wise loss function, that compares each pixel in the output to the corresponding pixel

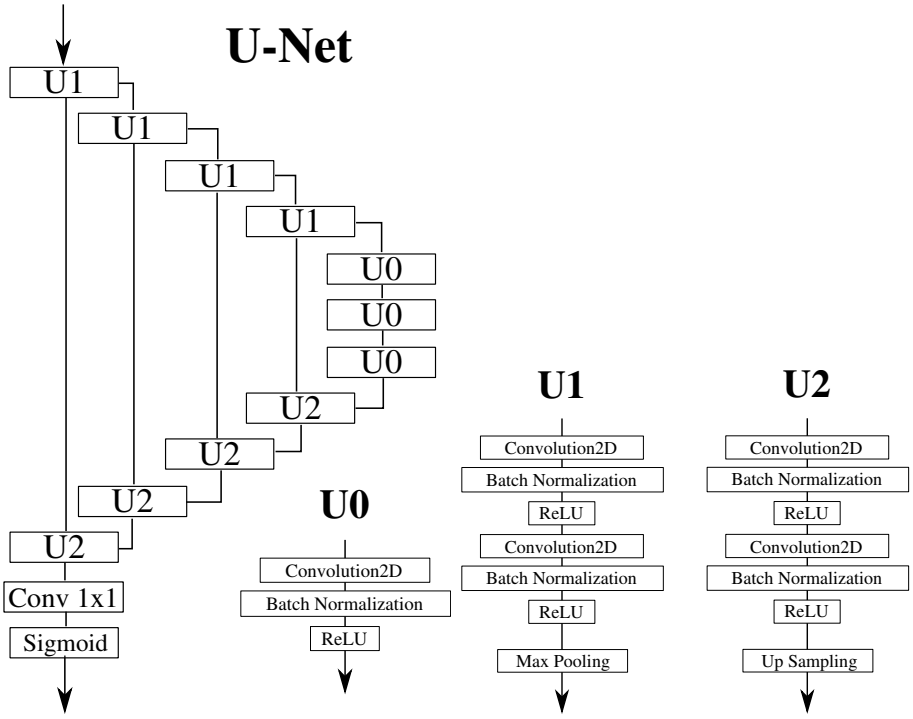


Figure 2.11: Network architecture schematic for the U-Net network (Ronneberger et al., 2015).

in some target image T, given some input image X. Letting X_{ij} represent the $(i,j)^{\text{th}}$ pixel in the image X, then we have

$$\mathcal{L}(X,T) = \sum_i \sum_j \ell(f(X;\mathbf{w})_{ij}, T_{ij}). \tag{2.48}$$

The power of Fully Convolutional Networks comes from the fact that no part of the architecture is dependent on the size of the input image. This means that a model trained on images of one size, can be applied to images of another size immediately and without modification. This is because we only learn relationships between local features rather than relationships based on the position of a pixel within the image.

2.5.5 Recurrent Networks

Another key concept in the application of neural networks is the recurrent neural network (RNN). Recurrent Neural Networks are most intuitively understood in the context of time series analysis, where a constantly updating model allows predictions of the next event without having to store all previous observations.

In their most simple sense, a recurrent network with input x_t is governed by the following equations.

$$h^{(t)} = \sigma\left(Uh^{(t-1)} + Wx^{(t)}\right) \tag{2.49}$$

$$o^{(t)} = Vh^{(t)} \tag{2.50}$$

$$y^{(t)} = \text{softmax}\left(o^{(t)}\right) \tag{2.51}$$

The input value at time t, given by $x^{(t)}$ is modified by some learned weight tensor W, the previous hidden state $h^{(t-1)}$ is modified by the weight tensor U and these results are concatenated to form the current hidden state value $h^{(t)}$, after transformation by some non-linearity σ. The output value $o^{(t)}$ is generated by transformation of the hidden state by the weight tensor V. A softmax operation then maps this output to a probability distribution $y^{(t)}$.

As has been observed by Pascanu *et al.* (Pascanu et al., 2012) and previously by Bengio *et al.* (Bengio et al., 1994), there are a number of issues surrounding training recurrent networks of this kind. Considering the loss at $t = t_n$

$$L(f(x^{(t)}, \mathbf{w}), y^{(t)}) = L(f(x^{(t)}, \mathbf{w}|\{x^{(0)}, \ldots, x^{(t-1)}\}), y^{(t)}), \tag{2.52}$$

we see that it depends on all prior values of x_t. If we consider 'unrolling' this time recurrence then we see that in order to compute the gradient of this loss we must pass backwards through the σ non-linearity in Equation 2.49. Both sigmoidal and tanh non-linearities have gradients close to zero almost everywhere so repeated composition of these functions can easily lead to vanishing gradients.

Work towards addressing this issue was proposed by Hochreiter *et al.* (Hochreiter, 1998; Hochreiter and Schmidhuber, 1997), with their Long Short-Term Memory (LSTM) network. This addresses this issue of vanishing and/or exploding gradients by introducing a 'memory' unit that is temporally connected, without having to pass through a non-linearity, allowing much longer range interactions to be learned.

$$f^{(t)} = \sigma_g(W_f x^{(t)} + U_f h^{(t-1)}) \tag{2.53}$$

$$i^{(t)} = \sigma_g(W_i x^{(t)} + U_i h^{(t-1)}) \tag{2.54}$$

$$o^{(t)} = \sigma_g(W_o x^{(t)} + U_o h^{(t-1)}) \tag{2.55}$$

$$c^{(t)} = f^{(t)} \circ c^{(t-1)} + i^{(t)} \circ \sigma_c(W_c x^{(t)} + U_c h^{(t-1)}) \tag{2.56}$$

$$h^{(t)} = o^{(t)} \circ \sigma_h(c^{(t)}) \tag{2.57}$$

$$y^{(t)} = \text{softmax}\left(h^{(t)}\right) \tag{2.58}$$

This differs from the original RNN formulation by the introduction of the 'memory unit' $c^{(t)}$, as well a system of 'gates' that will only expose the memory unit to the input or output data where appropriate. We see that while modifications to c require we pass through a non-linearity, information can pass through from $c^{(t-1)}$ to $c^{(t)}$ unimpeded. f acts as a 'forget' gate and can zero elements of $c^{(t)}$. $i^{(t)}$ represents an 'input' gate which modulates the effect both the current input and the previous output can modify the memory.

Although we have outlined networks here which are capable of emitting an output at each time-step, a common design pattern for recurrent networks is to accumulate knowledge about an entire sequence of data points before emitting a single summary output. A good example of this is for natural language processing (NLP) where recurrent networks have found great success in encoding the content of sentences for further analysis (Graves, 2013; Graves et al., 2013b; Sutskever et al., 2014). This represents one of the great strengths of recurrent networks in their ability to deal with inputs of indeterminate size, such as sentences, in a uniform way. In these discussions there has been a clear assumption of 'directionality' or an 'arrow of time', however, in some cases it may be advantageous for the recurrent network to be able to look backwards as well. An example of this would be a word towards the end of a sentence that provides context for one at the beginning of the sentence. In order to try and address this, some authors have tried applying LSTM units bidirectionally (Graves et al., 2013a; Assael et al., 2016; Chan et al.,

2015). This means training two separate units in parallel, which view data in opposite directions. The results from the backwards pass can then be reversed and concatenated with the forward pass to provide an additional set of output features for each time point, before further processing. However, it should be noted that not all users report improved results from this augmentation (Chung et al., 2016).

Recurrent Neural Networks have also been applied to static images. A novel approach, produced by Zheng *et al.* (Zheng et al., 2015), utilised a recurrent network to approximate the Mean Field Inference algorithm for dense CRFs of Krähenbühl and Koltun (Krähenbühl and Koltun, 2012). By formulating the iterative Mean Field algorithm as a recurrent network they were able to achieve something analogous to the popular CNN-CRF approach, as a fully end-to-end trainable system. A similar approach, although not explicitly rooted in the CRF, was taken by Pinheiro and Collobert (Pinheiro and Collobert, 2014), who formulated a Recurrent-CNN network for scene parsing.

2.5.6 Adversarial Networks

The final concept in deep learning that we will introduce is the adversarial training of neural networks. Initially proposed by Goodfellow *et al.* (Goodfellow et al., 2014) in their Generative Adversarial Networks (GAN) framework. They propose an 'adversarial' training paradigm, by which a generative model is taught to sample from the arbitrary distribution represented by a given training set. More generally, the framework attempts to learn a mapping from some tractable distribution, such as a multivariate Gaussian distribution, to some highly intractable distribution, representing the manifold that our training data exists on. Earlier work on Variational Auto-Encoders (VAE) has aimed to address this task (Kingma and Welling, 2013) although it is thought that pixel-wise loss functions are not sufficient for modelling many of the relationships required for generative networks. The Generative Adversarial Network (GAN) framework provides a novel approach to training generative networks without relying on pixel-wise loss functions. It is done by leveraging some auxiliary network, known as a discriminator network (D)

in order to train our generator network (G) for the desired purpose. The work by Goodfellow *et al.* formulates this relationship as a minimax game, where the optimal D and G pair satisfy the following relationship:

$$[D, G] = \min_G \max_D \mathbb{E}_{z \sim z_{syn}} \left[\log\left(1 - D(G(z))\right)\right] + \mathbb{E}_{x \sim x_{data}} \left[\log(D(x))\right]. \tag{2.59}$$

In this formulation, D gives an output in the range $(0, 1)$, attempting to discern whether its argument is 'real' or 'fake', 1 for fake and 0 for real. G is provided with an input from some 'seed' distribution z_{syn} and attempts to map it to the same distribution as the training data x_{data}. This can be broken down into two optimisation problems.

$$\mathcal{L}_D = -\left(\log(D(x)) + \log(1 - D(G(z)))\right), \tag{2.60}$$

$$\mathcal{L}_G = \log(1 - D(G(z))). \tag{2.61}$$

Informally, we train D to be able to discern real data from synthetic data, and then we train G to be able to deceive D with its synthesised data. Importantly, the generator network is never presented directly with the training data, meaning that it only improves by learning to deceive the discriminator network, rather than by memorising the training data.

Goodfellow *et al.* (Goodfellow et al., 2014) prove that this system has a unique global optima, where an ideal discriminator D^* is fooled by a perfect generator G^*, meaning $D^*(x) = D^*(G^*(z)) = 1/2$ for all x and z, leaving,

$$\mathbb{E}_{z \sim z_{syn}} \left[\log\left(1 - D^*(G^*(z))\right)\right] + \mathbb{E}_{x \sim x_{data}} \left[\log(D^*(x))\right] = -\log(4). \tag{2.62}$$

However, this adversarial training scheme can be difficult to train, largely attributed to the vanishing gradients of the KLD in the case that the distributions x_{data} and $G(z_{syn})$ are disjoint. A number of authors have attempted to address this. The Wasserstein GAN attempts to address this by replacing this KLD comparison with an Earth-Movers distance, whose gradient is strong everywhere (Arjovsky et al., 2017; Gulrajani et al., 2017). In addition to this, a number of heuristics

have been suggested in order to stabilise training in adversarial settings (Salimans et al., 2016; Shrivastava et al., 2016).

However, generative modelling is not the only application for adversarial training. A number of recent works have identified the power of using adversarial training as an auxiliary loss for supervised training, using conditional GANs (Mirza and Osindero, 2014). As mentioned earlier in Section 2.5.4, the majority of fully convolutional methods train against pixel-wise loss functions. Although a degree of dependency is communicated between variables due to weight sharing within the network, pixel-wise loss functions are incapable of encoding any kind of structure in the output and thus assumes independence between output values. In this context the discriminator network itself can be considered to be taking the roll of a (learned) loss function in order to model the higher order dependencies within the output. This has been successfully applied to image-to-image translation (Isola et al., 2016; Zhu et al., 2017), in-painting (Denton et al., 2016; Pathak et al., 2016) and semantic segmentation (Luc et al., 2016).

2.5.7 Literature

Although 2D CNNs are ubiquitous for 2D imaging tasks, the optimal choice of network for volumetric imaging is less clear. Here we will restrict our comments to those works which apply to volumetric deep learning, as this is the most relevant field to our contributions in this thesis. We will also focus mainly on works applied to medical imaging applications. Comprehensive reviews of the state-of-the-art in deep learning as applied to medical imaging have been produced by Litjens *et al.* (Litjens et al., 2017) and Shen *et al.* (Shen et al., 2017).

One of the causes of the divergence of approaches in volumetric deep learning is due to the fact that there is far more variation in what this additional dimension represents. In the case of brain MRI, it may simply represent an additional isotropic image dimension, in which case it would make little sense to differentiate this z dimension from the original x and y dimensions. Earlier work in data of this kind applied multiple orthogonal 2D networks in order to build a single unified

analysis of the voxel at the intersection of these planes. This approach has been applied to brain tumour segmentation by Lyksborg *et al.* (Lyksborg et al., 2015), to knee cartilage segmentation by Prasoon *et al.* (Prasoon et al., 2013) and to lymph node detection by Roth *et al.* (Roth et al., 2014). This application of a number of orthogonal, independent, 2D CNNs was largely driven by hardware limitations on GPU memory, with 3D applications often having significantly higher demands on memory consumption.

Shallow 3D CNNs have been applied to PET-MRI data completion by Li *et al.* (Li et al., 2014). Deep 3D CNNs have been applied successfully to brain lesion segmentation by Kamnitsas *et al.* (Kamnitsas et al., 2016b) with their Deep Medic architecture. In this work they utilise a two-streamed CNN network - a full resolution stream for learning textural features and a low resolution stream for learning location and other more general image properties. These parallel streams are combined before a final prediction is produced.

3D CNNs have also been explored in the V-Net architecture of Milletari *at al.* (Milletari et al., 2016) for prostate segmentation in MRI and the 3D extension of U-Net (Çiçek et al., 2016). In both the U-Net and V-Net architectures, a multi-scale analysis of the image is also considered, however, in these architectures they are not explicitly decoupled as in the Deep Medic architecture. In the work of Milletari *et al.* they also suggest the use of the Dice coefficient as a loss function for segmentation tasks, although this has been used elsewhere previously.

Ideas from the ResNet (He et al., 2016) have also been extended to 3D in the VoxResNet architecture (Chen et al., 2016a) applied to brain MRI. The ResNet architecture achieved state-of-the-art performance on the ImageNet2016 challenge by observing that it was possible to train extremely deep networks by treating the identity mapping as the default behaviour of a neural network unit, rather than the zero mapping. This is achieved by allowing skip connections over sets of layers before subtracting the results. This means that rather than learning $\hat{x} = f(x)$, we learn $\hat{x} = x - f(x)$, where $f(x)$ represents the mapping learnt by the intermediate layers. The architecture of Chen *et al.* applies this insight,

along with an auto-context configuration to achieve state-of-the-art results on the MICCAI MRBrainS Challenge.

In addition to convolutional networks, arrays of orthogonal recurrent networks have been applied to images and image volumes. These networks traverse an image along each of its dimensions in order to model the full system of dependencies within an image. This was pioneered by Graves and Schmidhuber for handwriting recognition (Graves and Schmidhuber, 2009) using their Multi-dimensional LSTM networks (MD-LSTM), but these methods suffer from poor complexity properties. This was extended to be applied to generalised structured data with the Grid-LSTM (Kalchbrenner et al., 2015) and to include residual connections in the Pixel-LSTM (van den Oord et al., 2016). This kind of network has also been applied to volumetric biomedical imagery by Stollenga *et al.* (Stollenga et al., 2015) with their PyraMiD-LSTM network. In this work they aim to improve the computational properties of the MD-LSTM by utilising the convolutional LSTM (ConvLSTM) and rearranging computations in order to better exploit GPU parallelism.

Alternate methods for handling volumetric data may become more appealing when our z dimension is distinct from the x and y dimensions. In these cases it may be desirable to treat our volumetric image as a sequence of images and this becomes the domain of recurrent-convolutional hybrid networks. In videos this sequence exists temporally and recurrent networks have been used to great effect in conjunction with CNNs (Graves, 2013; Assael et al., 2016; Chan et al., 2015). However, in medical imaging it is most likely that, while the dimensions are all spatial, this anisotropy is induced by the imaging modality itself. Recurrent networks have been incorporated into 2D CNNs by Poudel *et al.* (Poudel et al., 2017) for the task of cardiac segmentation in MRI, known for highly anisotropic voxels. In this method the recurrence is applied to the internal layers of a U-Net style network. This means that abstract, low-resolution information is shared between the image slices, but fine detailed considerations depend only on the current slice. The convolutional LSTM provides a natural way to model sequences of images by replacing the inner product operations of the LSTM with kernel

convolution operations. This was first introduced by Shi *et al.* (Shi et al., 2015) for weather forecasting, but has since been applied to biomedical imaging by Chen *et al.* (Chen et al., 2016b). In this work, Chen *et al.* combine a modified U-Net CNN network with a stacked ConvLSTM architecture in order to perform segmentation on anisotropic microscopy images. In this work they explore a number of stacked U-Net architectures which they term the kU-Net. They compare their preferred kU-Net, combined with a U-shaped ConvLSTM architecture to a number of state-of-the-art, volumetric networks and found favourable performance in both accuracy and memory usage on two public data sets.

2.6 Summary

New developments in the imaging of tumour micro-vasculature provide a fantastic opportunity for discovery in cancer research, however, they also present a number of technical difficulties for automated analysis. From this review of the literature we are able to identify a number of shortcomings in the existing work, for this particular task, which we will address in this thesis.

- Although superpixels have been applied to medical imaging volumes with some success, their original formulation relies on strong boundaries between objects of interest. This is often not the case in medical images. In order to address this we have developed a novel SLIC formulation, hesSLIC, that transforms the image domain into one more amenable for SLIC superpixel clustering on medical images. This has been specifically designed for curvature dominated structures, but we believe it to be generally applicable to medical images.

- Markov Random Fields have been heavily used in medical imaging for some time, but their application to superpixels has not been fully explored, specifically in vascular segmentation. We propose an anisotropic regularisation scheme for superpixels that incorporates a vesselness property.

- Work on vessel segmentation using developments in deep learning has focused almost entirely on 2D retinal fundus segmentation tasks. We will explore the use of fully convolutional networks in segmenting complex micro-vasculature without contrast agents. Current deep learning methods have also largely focused on segmentation. We will expand on this by learning some degree of 3D 'vesselness' from 2D considerations alone.

- The optimal use of 3D information in medical imaging is an open topic of discussion. Convolutional LSTMs have been shown to be an effective way to link together 2D CNN networks for highly anisotropic 3D images. We propose the use of extremely lightweight ConvLSTM architectures and we demonstrate the ability of these networks to extract 3D medial representations of the vascular structures, something that is not possible using only 2D networks.

- Uncertainty in the annotation of sparse structures such as medial/skeletal representations of vessels can make this a challenging task for supervised learning. We propose the use of adversarial training to learn to extract the centreline of vessels directly. This combines the traditional pipeline of segmentation and skeletonisation into a single end-to-end training process.

3

Segmenting Vasculature with a Supervoxel-MRF and Random Forests

Contents

3.1 Introduction 64
3.2 Related Work 66
3.3 Materials 68
3.3.1 Data 68
3.4 Methods 69
3.4.1 Supervoxels 69
3.4.2 Markov Random Fields 74
3.4.3 Random Forests 79
3.4.4 Annotation 80
3.5 Results 81
3.6 Discussion 87

3.1 Introduction

Segmentation of vasculature continues to be an important and challenging topic of research in biomedical image analysis. An area where the investigation of vasculature is of particular interest is in the study of tumour angiogenesis. Neovascularisation is known to be one of the key mechanisms in the growth and development of tumours (Hanahan and Weinberg, 2000). While the vasculature present in tumours can provide valuable information for investigating the properties of a tumour, there are various biological and technological issues surrounding the imaging of this vasculature that any automated segmentation method must address. Most methods for angiography use bloodpool based contrast agents, which provide contrast against the surrounding tissue for blood vessels imaged with modalities such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI) or Ultrasound (US). Bloodpool based contrast agents typically enhance vessels as bright, tubular objects, and many methods for vessel segmentation exploit this tubular structure through analysis of the local Hessian eigensystem (Rudyanto et al., 2014). However, in tumours, we know that the vasculature system can be poorly or irregularly perfused due to the inherent leakage and chaotic organisation (Weinberg, 2013). By imaging vessels using only bloodpool based contrast agents, we would implicitly limit our analysis to only the perfused vasculature. Therefore, as we are interested in both well perfused and poorly perfused vasculature, any such contrast enhancement-based segmentation method will be insufficient.

Fluorescence microscopy provides an alternative way to image vasculature, by imaging the vascular endothelium directly. This can be done pre-clinically by causing the endothelial cells to express a fluorescent protein, which can then be directly imaged. However, particularly in the case of tumour vasculature, the endothelium alone does not provide the tubular geometry that most standard methods of vascular segmentation rely upon, as only the outer wall is visible. In the imaging paradigm of bloodpool based contrast agents, methods typically attempt to infer the presence of vasculature from the perfusion. This is fundamentally flawed in the case of leaky and poorly perfused vasculature, and therefore we propose a much more robust

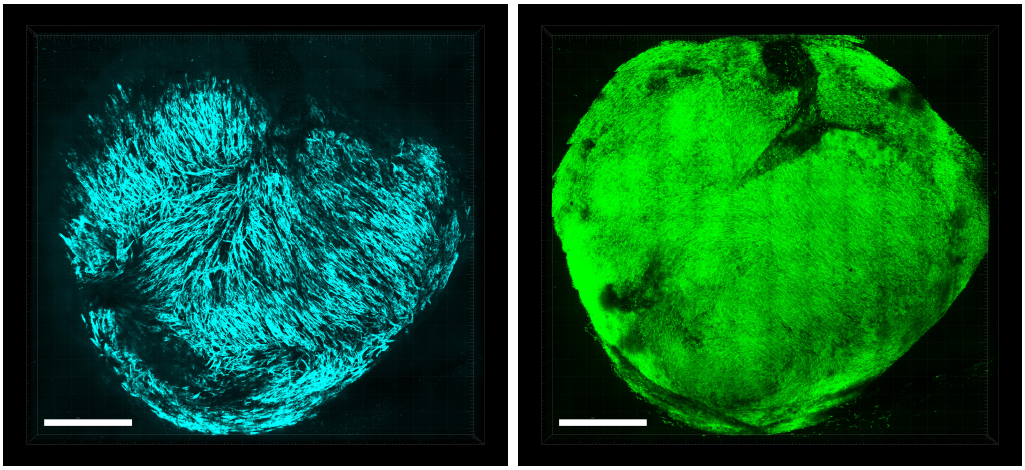


Figure 3.1: Whole tumour images (225 tiles) showing endothelial cells (cyan) and the corresponding tumour cells (green).

paradigm, to infer the perfused as well as non-perfused lumen of the vessel from the visible vascular walls, which will always be present, without imaging the lumen directly. For our proposed segmentation algorithm we use signals from both the endothelium and tumour cells fluorescence channels, avoiding any kind of bloodpool labelling. A whole tumour view of these channels can be seen in Figure 3.1 and single images tiles can be seen in Figure 3.2. Further details regarding these images and the acquisition protocol is provided in Section 2.1.1.

In addition to these biological concerns, modern microscopy imaging techniques are capable of producing extremely large image volumes. While this allows us to image small structures with high resolution, there are challenges in terms of processing the entire volume in an efficient manner. Methods designed to operate on large image volumes such as these must have reasonable bounds on both computational time and memory usage.

To address these difficulties, we propose a method of segmentation in which we first perform a supervoxel segmentation of the image volume. We then extract image features from each supervoxel which are classified using a Random Forest classifier to determine the most likely structure present in the image. These likelihoods are then regularised by a vessel-oriented Markov Random Field (MRF) graphical model defined over the graph of supervoxels to generate our final segmentation of the vasculature. Central to our application is the ability to generate meaningful

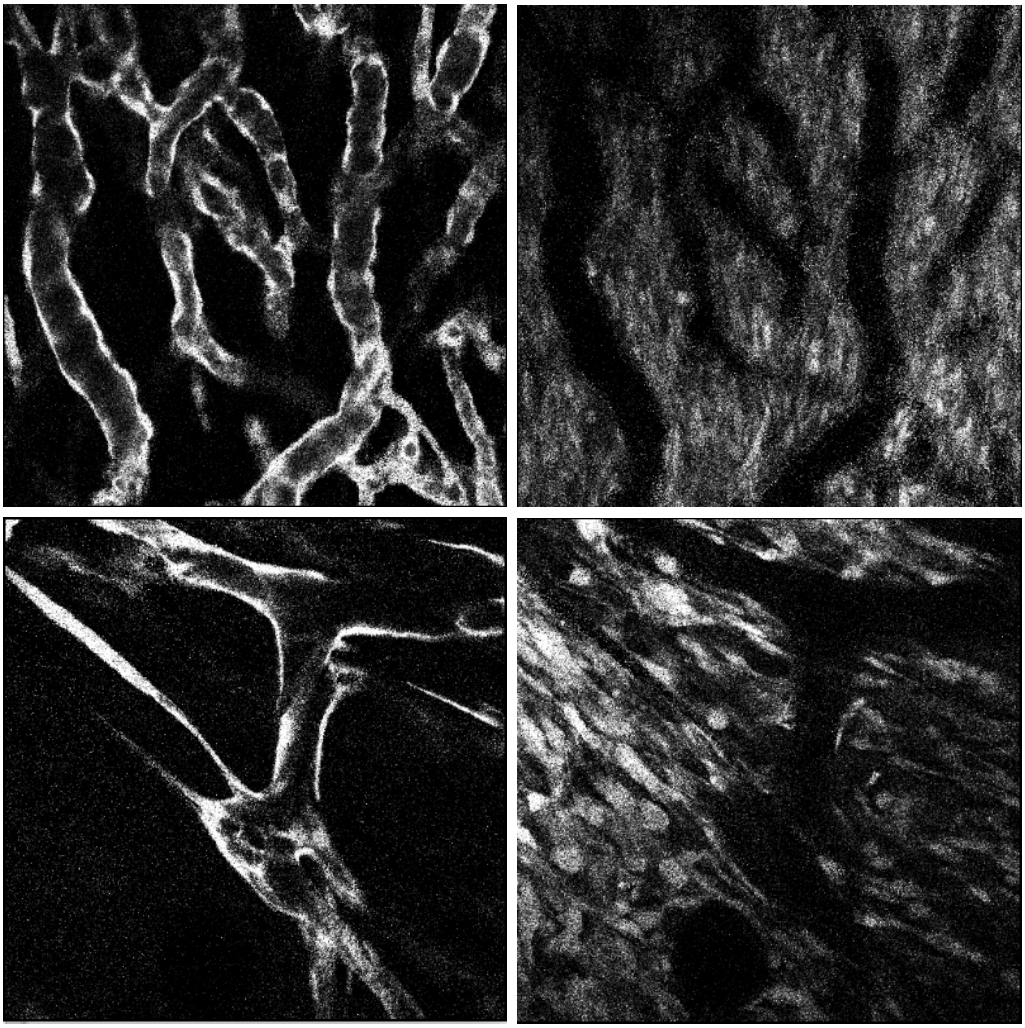


Figure 3.2: Fluorescence channels showing a single tile (512×512 pixels) of endothelial cells (left) and the corresponding tumour cells (right).

measurements of the underlying vasculature, rather than merely achieving a high volume overlap. We will show that not only does this method produce accurate and complete segmentations of the image, but that the semantic structure of the vasculature is maintained, allowing a straightforward skeletonisation process to recover the full topology of the network.

3.2 Related Work

There has been very little work focused on the segmentation of vasculature labelled purely on the endothelium. Instead, the vast majority of vascular segmentation techniques depend on the use of bloodpool based contrast agents, providing a

tubular structure (Frangi et al., 1998; Xiao et al., 2011; Krissian et al., 1998; Lorenz et al., 1997). A thorough review of methods applied to enhanced vasculature has been produced by Lesage *et al.* (Lesage et al., 2009). In fact, the lack of suitable segmentation techniques for data of this type means that even state-of-the-art imaging studies rely upon manual or semi-automatic segmentations to acquire measurements (Huang et al., 2014; Barber et al., 2003; Bentley et al., 2014; Meijer et al., 2016). In the case of many biological studies, manual annotation becomes infeasible due to the scale of images acquired. A full exploration of the spatio-temporal behaviour of tumour vasculature requires that we analyse many large image volumes, both longitudinally and for a number of cases. Therefore, a suitable automated method must be developed for data of this type. The method of Narayanaswamy *et al.* (Narayanaswamy et al., 2010) attempts to tackle a similar application by generating a bounding mesh from the endothelial cells. However, they focus on larger, non-pathological vasculature which does not suffer from the problems associated with the chaotic tumour neovasculature. Work by Almasi *et al.* on a similar domain avoids the general task of segmentation by enhancing 'key points' and utilising graph considerations to extract the network connectivity (Almasi et al., 2015, 2016). Deep Convolutional Neural Networks have been applied to a related problem also by Teikari *et al.* (Teikari et al., 2016). In this work they apply the ZNN framework of Lee *et al.* (Lee et al., 2015).

In the case of such large image volumes it becomes impractical to deal with each pixel individually, and a suitable reduced representation may be required. Often there exists a large amount of local redundancy in an image, so we may consider a piecewise constant representation of the image in some feature space. We refer to these regions as supervoxels, which have been used to great effect in many imaging applications. A number of supervoxel clustering algorithms have been proposed. Felzenszwalb poses it as a graph partitioning problem (Felzenszwalb and Huttenlocher, 2004), the Normalized Cut method of Shi and Malik solves a generalised eigenvalue problem (Shi and Malik, 2000) and the classical watersheds algorithm has also been used. Achanta *et al.* proposed the use of a localised k-means

clustering for their supervoxel algorithm, Simple Linear Iterative Clustering (SLIC) (Achanta et al., 2012) and a number of variants have been explored by transforming the clustering domain. Irving *et al.* performed SLIC clustering on the PCA modes of DCE-MRI signals (Irving et al., 2014a, 2016), Oda *et al.* suggest the inclusion of Hessian based signals into a SLIC algorithm (Oda et al., 2017) and Holzer *et al.* proposed a signal transform using the monogenic phase to provide a more suitable clustering domain for medical images (Holzer and Donner, 2014).

The use of supervoxel-based features in graphical models has been successfully applied to object recognition tasks by Fulkerson *et al.* (Fulkerson et al., 2009). While Fulkerson *et al.* enforces supervoxel label continuity as a hard constraint, the use of supervoxels as a soft label constraint was explored by Kohli *et al.* in their Robust P^n method (Kohli et al., 2009). In biomedical applications, supervoxel graphical models have been explored by Lucchi *et al.* (Lucchi et al., 2012) for the segmentation of mitochondria from SSEM volumes. In their work both shape and textural features are extracted from each supervoxel and are used to train an SVM classifier, which is then used in a CRF framework. The work of Zhu *et al.* (Zhu et al., 2013, 2011) has applied supervoxels to the segmentation of 3-dimensional vasculature. In this work they consider a semi-supervised graph transduction approach using Gaussian affinities. However, their use of bloodpool labeling meant that simple intensity and orientation features were sufficient for segmentation.

3.3 Materials

3.3.1 Data

For the experiments in this chapter we use both real and synthetic data as outlined in Section 2.1. For the experiments on real data we use 8 manually annotated image tiles of size $512 \times 512 \times 70$, with anisotropic voxel sizes $0.8\mu m \times 0.8\mu m \times 5\mu m$. All images are used for both training and testing in an 8-fold cross-validation framework. For synthetic data we use 10 volumes for training and 10 volumes for testing. For Dice-based comparisons we use the segmentation targets and for centreline based comparisons we use the skeleton targets.

3.4 Methods

Here we will give an overview of the proposed pipeline, followed by a detailed description of each stage. We begin by performing a transformation of our input image into our proposed hesSLIC domain. This domain is then used for SLIC local clustering to produce a superpixel over-segmentation. Features are then extracted from the entire original image and assigned to their respective superpixel. Aggregations and averages of the features are taken for each superpixel. These are then passed to a Random Forest decision tree ensemble, which defines a unary potential for our MRF graphical model. The pairwise terms are also defined between each pair of neighbouring superpixels according to various constraints. This system is then solved to find the most likely labelling of our image which is then taken as the final segmentation.

3.4.1 Supervoxels

The size of the images that may now be routinely acquired with microscopy methods provides a challenge for image processing, and for graphical models in particular. To address this, we first perform a pre-processing step by which we subdivide the image into a set of congruent regions known as supervoxels (Achanta et al., 2012). These supervoxels provide a piecewise constant, reduced representation of the image in some feature space. Defining a graph of the adjacency of these supervoxel regions is a much less demanding task and opens up the possibility of efficiently using probabilistic graphical models on our large image volumes.

In this work we consider the Simple Linear Iterative Clustering (SLIC) supervoxel method of Achanta *et al.* (Achanta et al., 2012). SLIC supervoxels are both computationally fast and memory efficient to calculate. They are known to provide regularly sized, uniform, supervoxels. The number and compactness of these supervoxels is controlled by the user. The SLIC method is an extension of k-means clustering from a feature space to a combined spatio-feature space. We begin by

initialising a regular grid of cluster centres with step size S. Then we assign each pixel to its closest center according to some weighted Euclidean metric:

$$Dist = \sqrt{Dist_F^2 + \left(\frac{Dist_S}{S}\right)^2 m^2} \tag{3.1}$$

where $Dist_S$ is the Euclidean distance between the center and the pixel, and $Dist_F$ is the Euclidean distance in feature space. The SLIC algorithm can be considered to exist on a sliding scale between k-means clustering and a regular grid. The position on this sliding scale is determined by the compactness parameter m. By tuning m, a balance can be found between edge localisation and regularity.

However, we find that the concept of simply clustering pixel intensities is not well suited to images such as those in our primary application, where there is not a clear distinctive edge between many of the different structures, and edges which do exist may be soft. This has also been observed by Holzer *et al.* , inspiring their monoSLIC method (Holzer and Donner, 2014), which utilises the monogenic phase of the image.

In order to address this we introduce a modified version of the SLIC algorithm, which we title hesSLIC. SLIC relies on strong image gradients to create image boundaries; however in the case of medical images, specifically of vasculature, strong boundaries are not present and instead, soft gradients show us the edge of objects, and curvature cues tell us about possible vascular structures. When attempting to cluster on the smoothed signal, there is no clear boundary between the regions of high and low signal, this can result in a 'banding' of supervoxels along this transitional region. In the transformed hesSLIC signal on the bottom, the boundaries are made clear and cluster boundaries will align with the signal transition boundaries. In order to demonstrate this, we observe that a step function, convolved with a smoothing kernel, creates a smooth, sigmoidal function. However, the original stepping point is recoverable by examining the sign of the second derivative. We perform a transformation of the pixel domain by first computing the eigenvalues $\lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ of the Hessian matrix H at each pixel in the image I, at scale σ. For these experiments we take $\sigma = 2$.

We define,

$$m = \arg\max_i(|\lambda_i|) \tag{3.2}$$

and

$$\phi(\lambda_m) = \text{sgn}(\lambda_m)(1 - \exp(-|\lambda_m \nabla I|)) \tag{3.3}$$

such that this switching behaviour is emphasised in the low-curvature transitional areas, rather than noisy regions in which the sign of the second derivative may oscillate wildly. While the monoSLIC method (Holzer and Donner, 2014) analyses the sign of the monogenic signal, our method considers the local curvature.

As the Hessian matrix is symmetric, our set of eigenvectors will form an orthonormal basis and the infinity norm of the eigenvalues gives the magnitude of the principal curvature. If we consider the eigenvalues to vary smoothly over the image domain, then in order for the largest magnitude eigenvalue to change sign we must transition through a region of zero curvature. This means we should be transitioning from 'convex' to 'concave' regions. Thus, by examining only the dominant eigenvalue we can partition the space into regions of consistent curvature, concave or convex. Here we create a strong boundary at points at which the dominant curvature changes direction. This means that the enforced supervoxel boundaries will exist along a subset of the so-called 'parabolic lines' of the surface, the set of points at which at least one of the principal curvatures is equal to zero. In the case of vasculature, we are sure that the principal direction, corresponding to the greatest curvature, is perpendicular to the direction of the vessel and we therefore create strong boundaries along the length of the vessel, as desired. In addition to this we normalise according to the local image gradient and mean curvature, which helps to prevent rapid oscillations of the signal in low signal regions. The transformed signal, ϕ in Equation 3.3, is to be used in the original SLIC algorithm with the $Dist_F$ term from Equation 3.1 being calculated on this signal rather than the image pixel values directly.

In Figure 3.3, we see the original noisy step function in the top graph. This signal is smoothed to eliminate noise and produce the middle graph. It is on this smoothed

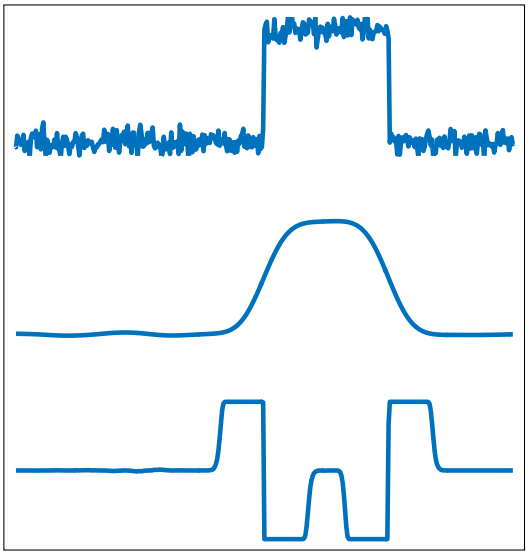


Figure 3.3: Top: Original noisy signal. Middle: Signal smoothed to eliminate noise (regular SLIC). Bottom: Signal transformed according to Eq. 3.3, recovering the hard edges.

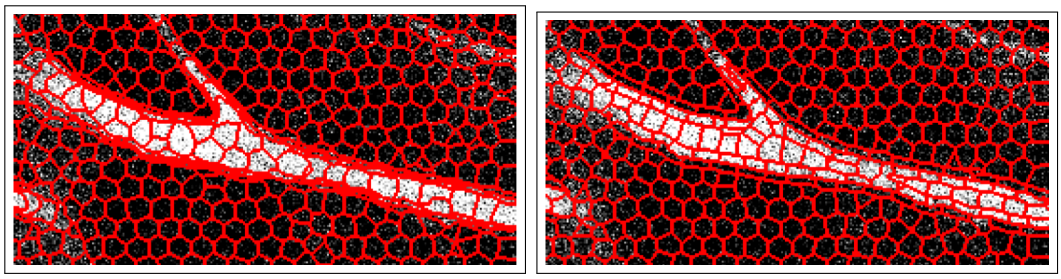


Figure 3.4: Over-segmentation of vascular structure by two different supervoxel algorithms. Top: original SLIC algorithm, banding occurs along edges. Bottom: hesSLIC signal transform has significantly reduced banding.

signal that the SLIC clustering would usually be performed. Once we transform the signal according to Eq. 3.3 we get the bottom graph. When attempting to cluster on the smoothed signal, there is no clear boundary between the regions of high and low signal, this can result in 'banding' along this transitional region. In the transformed hesSLIC signal on the bottom, the boundaries are made clear and cluster boundaries will align with the signal transition boundaries.

The results of this method may be observed in Figure 3.4 where the SLIC algorithm (top) suffers from the 'banding' issues described earlier, whereas the hesSLIC (bottom) manages to deal with this boundary appropriately. We find that on our data the hesSLIC method provides an excellent semantic deconstruction of

the image into structures such as lumen, endothelium, tumour and background.

An algorithmic overview of this method can be seen in Algorithm 4. We tune the number of supervoxels such that the size of a single supervoxel is roughly the size of the smallest structure in the image to be segmented.

Algorithm 4 Overview of hesSLIC algorithm

Require: Image I
1: Initialise output image ϕ
2: **for** pixel p in image I **do**
3: Compute local Hessian matrix at pixel p in image I
4: Compute magnitude of local image gradient at pixel p in image I
5: Compute eigenvalues λ of Hessian matrix
6: $m = \arg\max_i |\lambda|$
7: $\phi(p) = \text{sgn}(\lambda_m)(1 - \exp(-|\lambda_m \nabla I|))$
8: **end for**
9: Perform SLIC (Achanta et al., 2012) on new image ϕ.

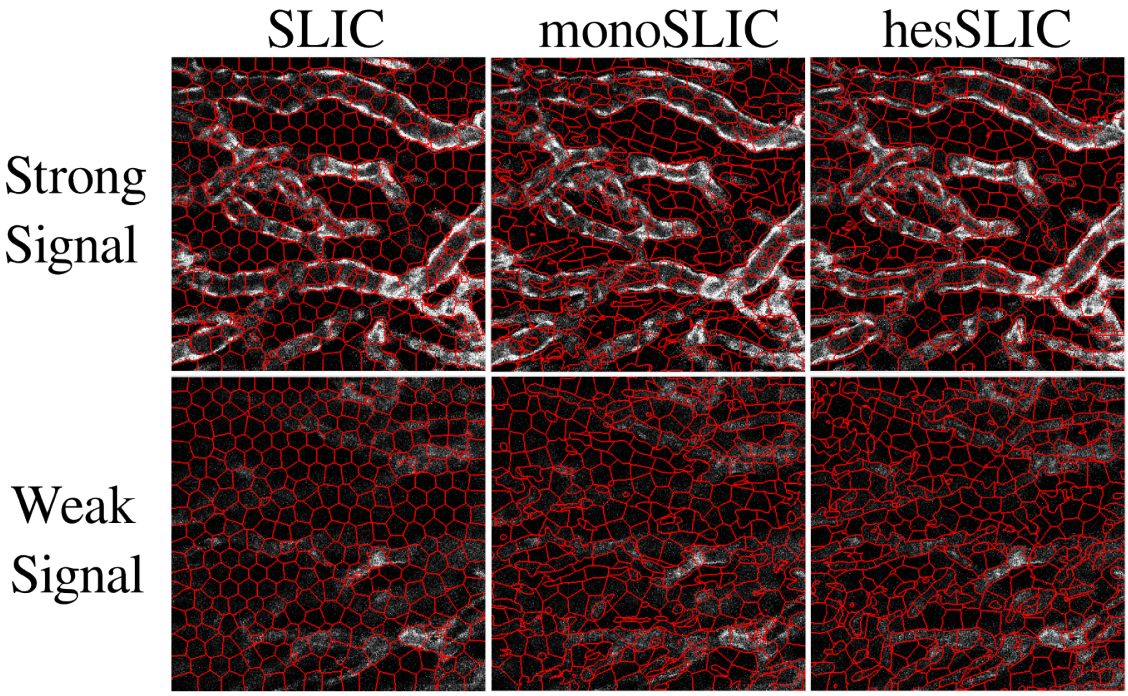


Figure 3.5: hesSLIC as compared to monoSLIC and regular SLIC. Left: SLIC, center: monoSLIC, right: hesSLIC.

By using the local curvature regime as the primitive for segmentation, we are able to produce a regular segmentation even in regions of extremely low contrast. This is highly desirable for microscopy images as these are known to suffer from loss of

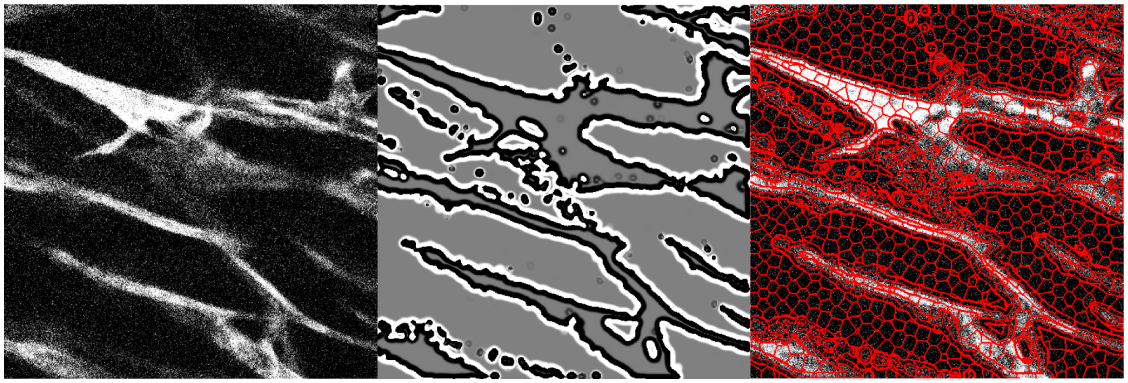


Figure 3.6: Transformation from endothelium channel (left) to ϕ domain (center) to superpixel segmentation (right).

signal as we increase penetration depth. As can be seen in Figure 3.5 the traditional SLIC method of Achanta clearly fails in the regions of low contrast. The contrast independent, phase-based method monoSLIC provides clear improvements however, our hesSLIC method appears to provide the clearest segmentation. We attribute this to the fact that the 2nd derivative domain is the minimal space in which we are able to describe vascular structures. The fact that we have used curvature cues means that this is possible even in regions of low signal intensity and in the presence of soft edges. An example of an input image, the subsequent signal transform (ϕ, Equation 3.3) and the resultant hesSLIC superpixel segmentation is shown in Figure 3.6.

3.4.2 Markov Random Fields

Markov Random Fields (MRFs) provide a principled framework in which we are able to combine probabilistic information, provided by the image, with a set of spatial prior beliefs. We consider this a semantic segmentation task where we attempt to assign a label to every part of the image, denoting what kind of structure is present at that particular location. In these images we identify 4 labels: tumour, endothelial cells, vessel lumen and background. This is how we motivate a 4-label pairwise Markov Random Field model.

In full generality, an MRF describes a joint probability structure over an undirected graph. If we combine this with the positivity condition, $P(\mathbf{x}) > 0, \qquad \forall \mathbf{x}$. It may be shown, by the Hammersley-Clifford theorem, that this is equivalent to

a Gibbs random field and as such can be factorised into conditional probabilities over the cliques c.

$$P(\mathbf{x}) = \frac{1}{Z}\exp\left(-E(\mathbf{x})\right) = \frac{1}{Z}\exp\left(-\sum_{c} E_c(\mathbf{x}_c)\right) \tag{3.4}$$

For tractability we consider only pairwise cliques,

$$E(\mathbf{x}) = \sum_{i \in N} U_i(x_i) + \sum_{(i,j) \in C} V_{ij}(x_i, x_j) \tag{3.5}$$

where E represents a potential induced by a given labelling x and Z is a partition function required for a correctly normalised distribution.

We can interpret this formulation as the function U_i defining the node i's inclination towards being assigned the label x_i. This function can be unique for each node i, since the function U_i is conditioned on the data itself. The function V_{ij} describes the interaction at the edge connecting node i with node j, taking labels x_i and x_j respectively. By penalising boundaries between different labels, we can provide a form of spatial regularisation. As our function V_{ij} may also be conditioned on the data it improves clarity to rewrite our energy as,

$$E(\mathbf{x}) = \sum_{i \in N} U_i(x_i) + \sum_{(i,j) \in C} G(f_i, f_j) H(x_i, x_j) \tag{3.6}$$

where, G is a function of the features at nodes i and j and H is a function of the labels alone. The functions G and H can then be defined globally. We can then solve this approximately using the Graph Cut method of Boykov *et al.* (Boykov et al., 2001).

As we reformulated our joint probability in terms of a Gibbs random field in Equation 3.4 we can see that the maximum a posteriori (MAP) labelling will correspond to the labelling that minimises the energy E. As the partition function Z does not depend on the labelling $\mathbf{x}$ we can avoid computing it entirely. The problem of optimal labelling thus reduces to minimising an energy of the form given in Equation 3.5.

In general, finding this optimal labelling is an NP-hard problem. However, the Graph Cuts methods (Boykov et al., 2001) provide us with a family of move-making algorithms for which we can obtain good optimality guarantees, provided that we

restrict our pairwise term V_{ij} to the set of *submodular* functions over the label space. The reason for this is that the graph cuts method relies on the min-cut/max-flow duality over an *s-t* graph (Boykov et al., 2001). Submodularity of the pairwise interactions between each label is a necessary condition for the strong duality of these problems to hold. As we separated the function $V_{ij}(x_i, x_j) = G(f_i, f_j)H(x_i, x_j)$ and only H is a function of the labels, we simply require that H is submodular and that $G \geq 0$ to enforce submodularity for the entire problem. For this application H will take the form of a Potts model and thus sub-modularity is enforced.

In our formulation we treat each superpixel as a node of the graph. Here, f_i, p_i and S_i correspond, respectively, to the feature vector, centroid location and set of constituent pixels for superpixel i. Our unary potentials are taken as the probabilistic outputs from a Random Forest classifier (Breiman, 2001), the details of which will be addressed in Section 3.4.3. As we wish to define a potential rather than a probability, we denote the probabilistic output of the Random Forest U_{RF} and that

$$U = -\log(U_{RF}). \tag{3.7}$$

We define our pairwise function over the features, $G(f_i, f_j)$, to be the product of four functions. Through the function G_1, we enforce that the strength of interactions is modulated by the distance between two nodes. The function G_2 enforces that interactions are strongest between nodes with similar features. The difference in size between the supervoxels is accounted for by G_3 and the agreement of the local directionality of the two nodes is dealt with by G_4.

$$G_1 = \exp\left(-\frac{|p_i - p_j|^2}{\gamma_d}\right) \tag{3.8}$$

$$G_2 = \exp\left(-\frac{|f_i - f_j|^2}{\gamma_f}\right) \tag{3.9}$$

$$G_3 = \frac{|i|}{\min(|j|, N_{min})} \tag{3.10}$$

$$S_{i,j} = (v_i \cdot \tfrac{p_i - p_j}{||p_i - p_j||}) \tag{3.11}$$

$$G_4 = |S_{i,j}| - (|S_{i,j}| - 1)\exp\left(-\frac{f_{i,v}}{\gamma}\right) \tag{3.12}$$

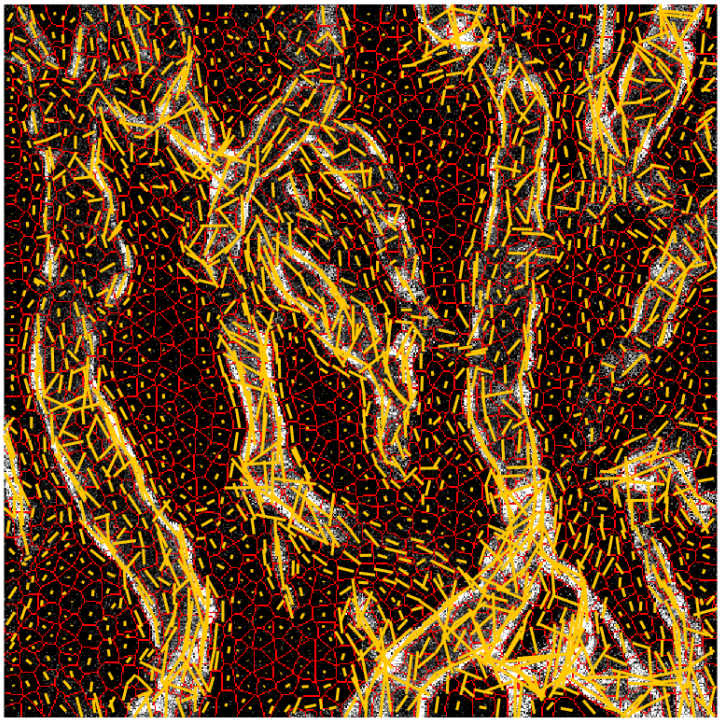


Figure 3.7: Superpixels overlaid with estimated directional vectors v and magnitudes $f_{i,v}$, used in the calculation of G_4 in Equation 3.12.

$$G(i,j) = \lambda G_1(i,j)G_2(i,j)G_3(i,j)G_4(i,j) \tag{3.13}$$

Tuning parameters γ_f,γ_d and γ govern the strength of these interactions and can be either calibrated or estimated from the data. $f_{i,v}$ is given by the local Hessian strength (L_2-norm of the eigenvalues), however, this could be replaced by a vesselness measure for suitable data. This term modulates the extent to which anisotropic regularisation should occur. The general strength of regularisation is modified by the parameter λ.

We define H to represent our biological understanding of the image, e.g. that lumen should not co-occur with tumour cells (because it should be surrounded by endothelial cells) or that endothelium should not share a boundary with background (because they should be embedded in the tumour tissue). Since, for the purposes of evaluation, we are comparing binary segmentation methods, we represent H by a simple Potts model, taking a constant value for different labels and 0 if

the labels are the same. However, more complex relationships between labels may be learned and encoded.

The combination of G_1 and G_3 means that we can avoid the issues of a non-regular supervoxel lattice, by taking a fully connected graph and eliminating sufficiently weak interactions. G_1 allows us to set a characteristic interaction distance by γ_d and G_3 prevents undue influence from small supervoxels. The interaction term G_2 provides us with the classical edge preserving regularisation that is responsible for the majority of the regularisation. However, it is known that regularisation of this form can suffer from the so-called 'shrinking problem' where larger, convex structures are encouraged in order to minimise boundary areas. Anisotropic regularisation terms such as $S_{i,j}$ have been suggested in terms of MRF formulations previously (Kaynig et al., 2010, 2015; Grau et al., 2006) however we incorporate a transition function G_4 to give highly anisotropic interactions only in regions of strong curvature ($f_{i,v}$) and isotropic interactions elsewhere. This takes a form similar to that found in the Vessel Enhancing Diffusion of Manniesing *et al.* (Manniesing et al., 2006b). In Figure 3.7 we show the the estimated directionality and magnitude for each supervoxel, denoted by v_i and $f_{i,v}$ in Equation 3.11 and Equation 3.12, respectively.

A key criticism of anisotropic regularisation in this domain relates to the fact that vessels only possess a tubular appearance along the length of a branch, and not at branch points. We believe that this regularisation scheme addresses this concern by falling back to regular isotropic regularisation in regions of low 'vesselness'. Although we have attempted to encode the structure of anisotropic vascular networks into these regularisation terms, it should be acknowledged that 1st order MRF schemes represent only a fairly limited subspace of all possible regularisation functions. Although higher order graphical models may be more expressive, we then become hamstrung by more complex and computationally intensive optimisation procedures.

3.4.3 Random Forests

In order to produce desirable segmentations, we wish to incorporate expert knowledge into the segmentation algorithm. Unfortunately, densely annotated 3D volumes are extremely time consuming to acquire and there are no guarantees of such a data set existing for any particular task. To this end, we require a method where images can be efficiently and sparsely annotated. These annotations can then be used to train a supervised learning algorithm which can provide a local distribution of likelihoods for each label. These can then be fed back into the MRF framework as unary potentials. Supervoxels provide an excellent way to extend sparse annotations into denser annotations. In addition to leveraging them towards a larger number of training pixels, it is also possible to extract richer features such as local texture and intensity histograms from the supervoxel as a whole.

As we wish to develop a general segmentation strategy, it follows that our learning approach should be able to deal with a wide range of features, both in terms of number of features and potential scaling. We would also like a method that naturally extends to probabilistic multi-class classification. Although recently Convolutional Neural Networks (See Section 2.5.4) have become the current state-of-the-art in multi-class semantic image segmentation, they require large training sets of densely annotated examples. We desire a method that is able to deal with a variety of features naturally, without requiring extremely large training sets. For these reasons, Random Forests are the natural choice for our purposes. An in-depth discussion of Random Forests and decision forests more generally is provided in Section 2.5.1.

We perform Random Forest classification for each supervoxel by constructing a feature vector similar to that used by Kaynig *et al.* (Kaynig et al., 2010, 2015). This feature set involving smoothing the training data using Gaussian kernels taken at a number of scales. We then compute image gradients, Hessian eigenvalues and pixel values at each scale and consider various combinations of these values across scales. In addition to these isotropic features we also compute steerable features taken at various rotations as well as the minimal, maximal, mean, median and variance of responses for all orientations to create rotational invariance. This

feature set takes inspiration from the work of Dollár *et al.* (Dollár et al., 2006) in the supervised learning of edge detectors, that a large and varied feature set is optimal, allowing the machine learning algorithm to determine the useful features. This feature set contains approximately 150 features.

For each supervoxel, we then form a single feature vector by taking the median value achieved over the supervoxel for each feature. In addition to this, the feature vector for each supervoxel is concatenated with the average feature vector of its neighbouring supervoxels, providing additional context for the classifier as has been explored by Fulkerson *et al.* (Fulkerson et al., 2009). Features are computed on both the endothelium and tumour channels, as visible in Figure 3.2. For these experiments we trained a decision forest consisting of 100 trees, with a maximum tree depth of 10 splits. Values of σ for the Gaussian kernels used for feature extraction were taken between 1 and 20. On the advice of Breiman and Chen *et al.* (Breiman, 2001; Chen et al., 2004), for training Random Forests on imbalanced classes, we draw bootstrapped samples of equal size from all classes in the training data. Therefore each individual tree is trained on balanced data. We choose the sample size to be the size of the minority class. Additionally, at each splitting node we take a feature subset of the size of the square root of the total number of features.

3.4.4 Annotation

We have developed an easy to use GUI for the annotation stage of the algorithm. The user simply draws on slices of a training volume with the appropriate labels. These sparse annotations are then extended to their supervoxel support regions. As our supervoxels are computed in 3D we are able to extrapolate annotations to unseen slices without the risk of mislabelling. The ability to extrapolate extremely sparse annotations to much larger regions of the image, particularly out of plane, via supervoxels is one of the key advantages in using supervoxel clusterings. We are able to generate sufficient training examples in a matter of seconds on an image that would take up to an hour to annotate fully manually.

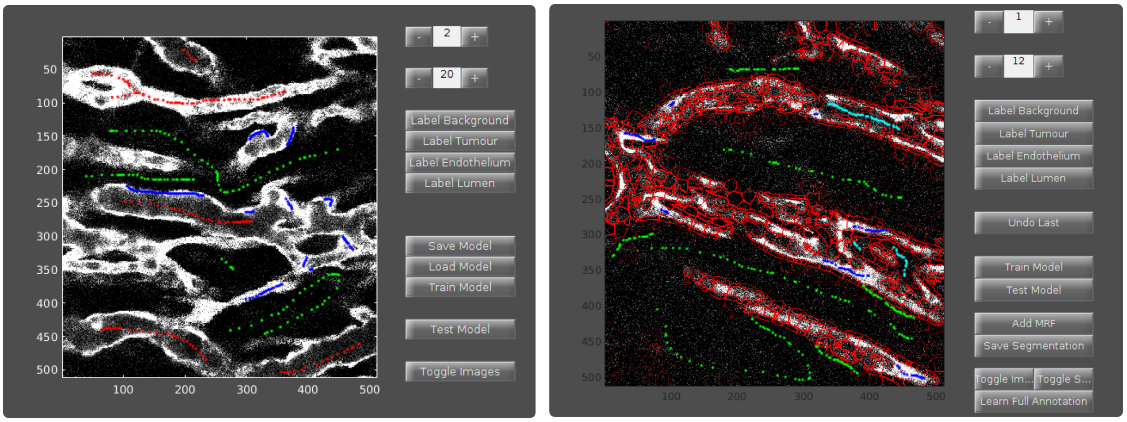


Figure 3.8: Left: Annotation GUI implemented in MATLAB with sample multi-label annotation 'scribbles' shown. Right: Uncertainty overlaid using superpixels to help guide the annotator towards more informative labels.

This approach also seamlessly lends itself to an active learning approach in terms of annotation. As the model can be quickly trained, an active learning cycle that utilises an entropy-based uncertainty labelling can significantly improve the efficiency of any annotation effort. Supervoxels also provide a useful overlay for displaying this uncertainty and make for a very natural annotation experience. An example of this is shown in Figure 3.8. Due to the relatively short training time, it is possible to freely incorporate model re-training into an annotation session. The user may therefore annotate some supervoxels, train the model, visualise the uncertainty overlay and focus annotations to the regions of higher uncertainty. We find this to be extremely effective and efficient. Here we calculate uncertainty as the prediction entropy, computed over all classes for each supervoxel:

$$H = -\sum_i p_i \log(p_i). \tag{3.14}$$

3.5 Results

We compare the results of our segmentation algorithm against a gold-standard, manual-segmentation produced for 8 image volumes of size $512 \times 512 \times 40$. We compared the manual segmentation against our method, a median-filtering approach, a commonly used Hessian-based, vesselness filtering method (Frangi et al., 1998) and a state-of-the-art vessel segmentation method specifically designed for fluorescently

labelled vessel laminae (Narayanaswamy et al., 2010). In the case of median filtering and vesselness, the images are thresholded to produce a final segmentation. The method of Narayanaswamy *et al.* was made available as part of the Farsight Toolkit[1] (Narayanaswamy et al., 2010). All the parameters and thresholds are optimised on a calibration image and held constant over the test set. The Random Forest model is trained over the entire data set in a leave-one-out framework, so for each test case, the model has been trained on all images, excluding the current test case.

We found, in the case of the method of Narayanaswamy *et al.*, that although the method had been specifically designed for data of this type, the computational time meant that this method would not be well suited for our purposes ($\sim$ 70000s per tile). This can be compared to our method, running at $\sim$ 100s per tile, and the median filtering method, at $\sim$ 10s per tile. We found that the method of Narayanaswamy performed well in regions of strong signal but failed to regularise in regions of low signal leaving incomplete sections of the vasculature as well as 'holes'. This is not unexpected as the method was designed for larger, more mature, vasculature, where the endothelium would be regular and more clearly labeled. The results of this method, in comparison with our method, can be seen in Figure 3.10.

We found the median filtering method to be by far the most effective of the other methods tested, due to its ability to 'fill' the lumen of the vessel by filtering the strong signal from the vessel walls. However, it is not a context aware method and we found that it was unable to fill the larger vessels found in our images. This can be observed in Figure 3.11.

We also investigate the role played by our proposed hesSLIC algorithm in improving segmentation. We perform this analysis on a synthetic dataset designed to replicate a number of the complexities of our real data, a description of this synthetic data is provided in Section 2.1.2. We compare the Dice overlap using both the traditional SLIC algorithm and our hesSLIC algorithm for two different values of the MRF regularisation. The results of this can be seen in Table 4.1, where we show the mean Dice score over our synthetic data set, along with standard

[1] www.farsight-toolkit.org

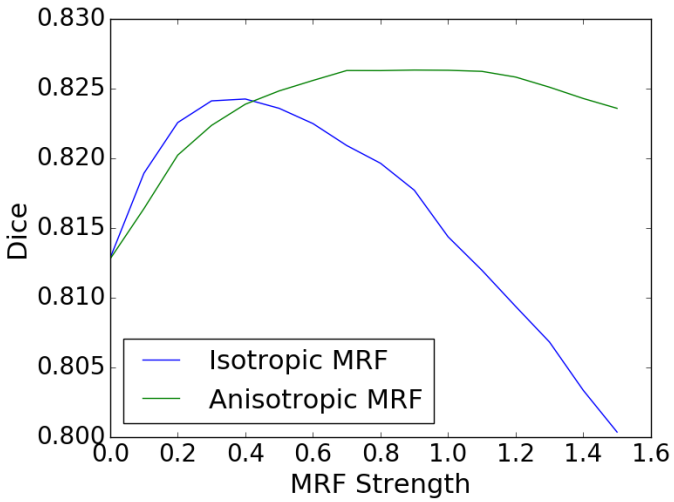


Figure 3.9: Comparison of the effects of MRF strength on isotropic and anisotropic regularisation.

deviations. Here the hesSLIC variant out-performs the regular SLIC algorithm both with and without MRF regularisation.

Method	Dice
SLIC-MRF00	0.77 ± 0.027
hesSLICMRF00	0.81 ± 0.025
SLIC-MRF05	0.78 ± 0.029
hesSLIC-MRF05	$\mathbf{0.83 \pm 0.027}$

Table 3.1: Results for hesSLIC vs SLIC comparison.

To explore the impact of anisotropic regularisation in the MRF, as imposed by Equation 3.12, we test the effectiveness on our synthetic data. Experiments were run over a range of MRF strengths, represented by the parameter λ in Eq. 3.13, both with and without the contribution from Equation 3.12. This provides a directional anisotropic regularisation, encouraging regularity in the direction of 'vessel structures'. We present an averaged Dice score over 10 synthetic image volumes. The results of this analysis are shown in Figure 3.9.

Although prevalent in the literature, we find the Dice overlap score does not tell the entire story with regards to the segmentation of vasculature. The Dice overlap relies on an assumption of local smoothness, such that changes in the score should be small with respect to small deviations from the optimal solution. This is not

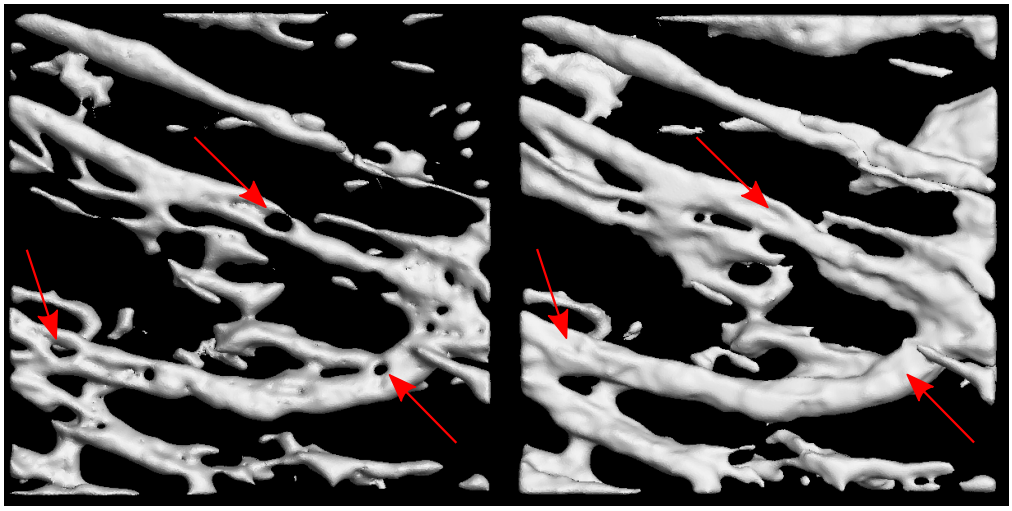


Figure 3.10: Left: Method of Narayanaswamy (Narayanaswamy et al., 2010). Right: Proposed Method. Here the method of Naranayaswamy struggles to fill the vessel lumina, leaving gaps in the segmentation.

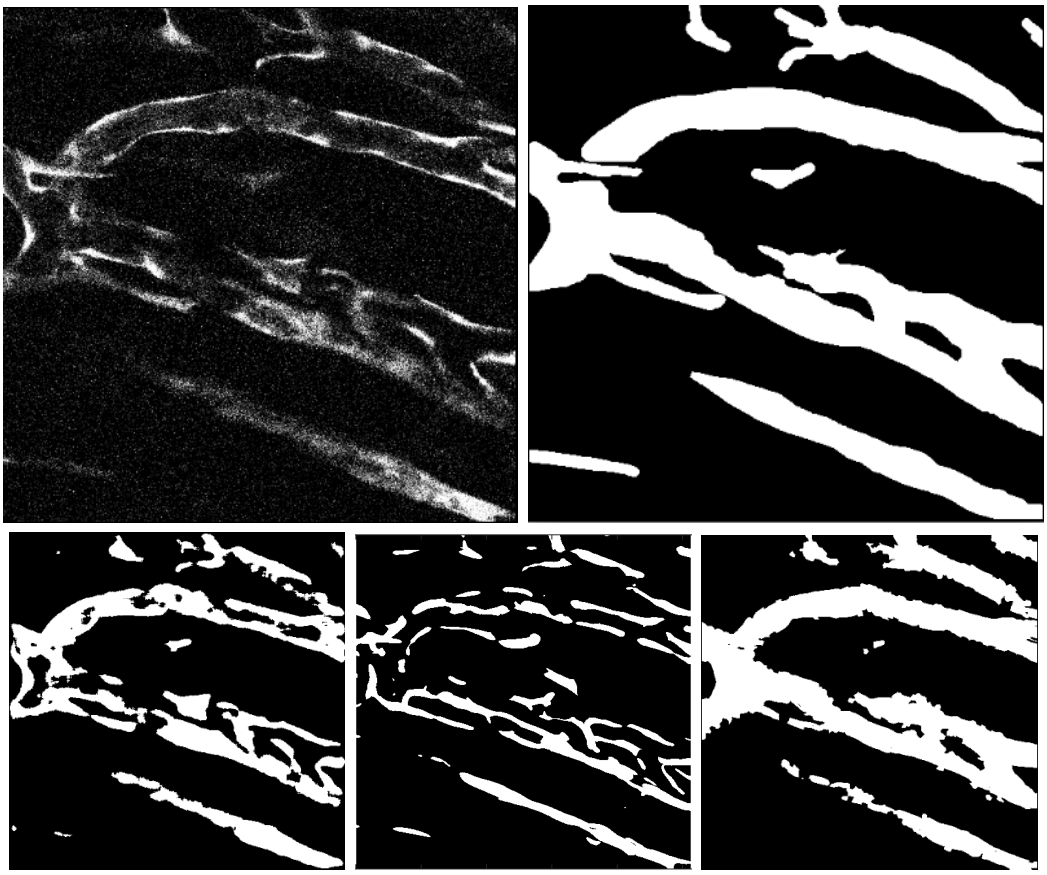


Figure 3.11: Top left: Endothelial cell channel, top right: manual segmentation, bottom left: median filtering method, botton centre: vesselness filtering method, bottom right: proposed method.

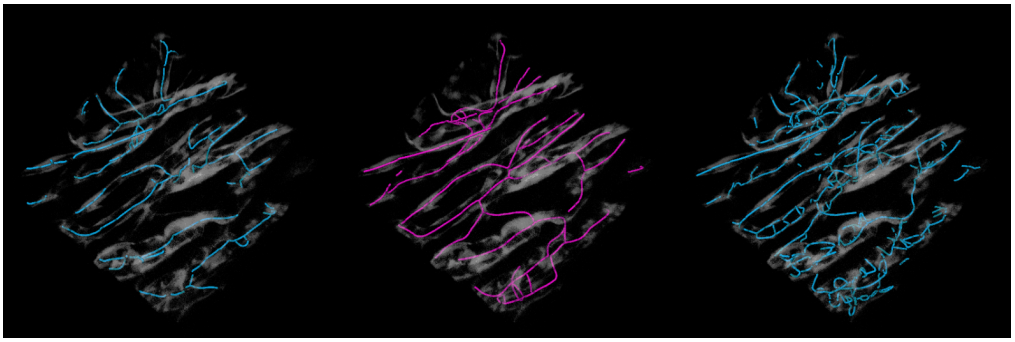


Figure 3.12: Skeletons extracted from segmentation. Left: Proposed method, Middle: Ground Truth, Right: Median Filtering.

the case in vasculature where small lateral changes perpendicular to the direction of vasculature may have a magnified effect on the Dice score. In addition to this, the Dice score does not assess the success with which a method has performed a true delineation, in this case that means its ability to recover the true topology of the vasculature. Although we report comparable results to the median filtering method in terms of Dice overlap, we also perform an analysis of the extracted vascular topology, via skeletonisation. In order to do this, we present the results of a Hausdorff analysis as well as the error in reported vessel diameter, this can be seen in Tables 3.2, 3.3 and 3.4. We derive a skeletal representation from each segmentation by applying a simple iterative thinning algorithm (Pudney, 1998). We refine each skeleton by pruning artefactual branches that did not exceed some minimum length, or the diameter of their parent vessel. For these experiments we chose a minimum branch length of 10 voxels.

As can be seen in Figure 3.12, although the two methods have a comparable volume overlap compared to the manual segmentations, the regularity of segmentation produced by the proposed method means a much cleaner skeleton is obtained, and thus a more accurate model of the vasculature. Disconnections and rough edges in the median filtered segmentation cause missing sections in the skeleton and many artefactual branches.

We define the Hausdorff Average (Dubuisson and Jain, 1994; Huttenlocher

Table 3.2: Dice overlap comparison

Case #	1	2	3	4	5	6	7	8
Proposed	0.789	**0.761**	**0.673**	0.679	**0.888**	**0.878**	**0.761**	**0.710**
Median Filter	**0.816**	0.665	0.599	**0.779**	0.841	0.739	0.575	0.368
Frangi Filter	0.353	0.326	0.310	0.297	0.295	0.250	0.232	0.134
Narayanaswamy	0.719	0.606	0.537	0.649	0.625	0.525	0.468	0.250

Table 3.3: Mean One-way Hausdorff Distance

Case #	1	2	3	4	5	6	7	8
Proposed	**5.02**	**11.66**	**11.94**	9.66	**4.96**	**4.61**	**4.85**	**4.01**
Median Filter	12.32	45.23	75.64	12.61	11.66	17.39	24.14	26.07
Frangi Filter	9.52	26.90	23.44	12.70	8.38	11.94	12.52	10.12
Narayanaswamy	7.48	12.69	13.89	**7.81**	6.78	6.78	8.06	9.22

Table 3.4: Mean Vessel Diameter Error

Case #	1	2	3	4	5	6	7	8
Proposed	**2.61**	**4.37**	**4.51**	4.72	**1.94**	**1.65**	**1.61**	**1.20**
Median Filter	3.72	5.69	6.34	**4.31**	2.55	2.91	2.61	2.00
Frangi Filter	3.43	5.07	5.63	4.58	4.58	4.28	3.16	3.66
Narayanaswamy	2.99	5.12	5.16	4.89	4.82	5.84	3.70	5.96

et al., 1993) as

$$d_H(A,B) = \frac{1}{|A|}\sum_{a\in A} d(a,B) \tag{3.15}$$

where $d(a,B)$ is the minimum Euclidean distance between a point a and a set B, given by $d(a,B) = \min_{b\in B}|a-b|$.

In this case the sets A and B denote the set of skeletal points of the segmented image and the ground truth image respectively. We propose this as a method to quantify the semantic value of the segmentation. While a Dice overlap shows that a significant pixel co-occurrence is achieved, the pixels that are not included in the segmentation may fundamentally change the meaning of the segmented structure. By quantifying the segmentation according to this distance metric rather than a simple segmentation overlap, we reward a method which captures the semantic meaning of the image, the vascular network.

We test all hypotheses using a Wilcoxon Signed-Rank Test, with Holm-Bonferroni correction for multiple comparisons. Details on these methods are provided in Appendix B. Although we observe no significant difference in the values of the Dice overlap, we also test the significance of the Mean Hausdorff Distance and Mean Vessel Diameter Error. We report a significant ($\alpha = 0.05$) improvement of the

proposed method over the others in both the Hausdorff average distance (Table 3.3) and the mean vessel diameter error (Table 3.4).

3.6 Discussion

Here we have presented a method for the multi-label segmentation of vasculature in 3D, endothelium-labelled, multi-photon microscopy images. We build up a primitive structure of the image using a novel supervoxel algorithm and generate likelihoods over the labels for each supervoxel, using a supervised Random Forest algorithm. These likelihoods are then regularised using a vessel-oriented MRF, which encodes ideas about the anisotropic nature of vasculature as well as the relative likelihood of co-occurrence for different labels. We have shown that this method out-performs a number of other methods, most notably in its ability to produce full segmentations of the vasculature, and thus accurately extract the underlying topology of the vasculature.

A key advantage of the method proposed here, in comparison to other segmentation methods, is that although here we compared against binary segmentations, the method extends to as many classes as are present in the training annotations. An example in this case would be to separate the image into tumour cells, lumen, endothelial cells and background, as mentioned previously. This can be done to an arbitrary degree with any biological understanding of the relationship between labels encoded in the label affinity function H from Eq. 3.6. In motivating this method, we placed great importance on the ability to fully capture the topology of the vascular network in an efficient manner. The ability of this method to achieve both of these goals allows the analysis of large, irregular, vascular structures such as those found in tumours during angiogenesis. Although many segmentation methods value a high Dice overlap, we emphasise the fact that any region identified by the segmentation method should correspond to a complete section of the vasculature. It is this desire that leads us to prioritise the Hausdorff distance. One of the key strengths of the proposed method is the ability to fully segment sections of the vasculature. However, this context awareness can cause the method to reject

sections of vasculature that are only partially visible. The ability of the thresholding method to segment only partially visible sections of the vasculature produces a favourable Dice score but makes the results wholly unsuitable for further analysis. This can be observed in Figure 3.12 where, although the median filtering method detects vasculature in approximately the correct parts of the image, the skeleton that is extracted is disjointed and contains many artefactual branches due to the incorrect topology of the segmentation.

From our synthetic experimental results, shown in Table 4.1, we see an improved ability to segment the structures when using the hesSLIC formulation. We intuitively attribute this to an ability to perform better clustering in the presence of soft gradients in the image. In our synthetic experiments into the effects of modifying MRF strength λ, shown in Figure 3.9, we also see an improved result by the application of anisotropic regularisation. In addition to a higher peak, we note that the results in the anisotropic case appear less sensitive to the value of λ. This could potentially be due to a reduction in the impact of the 'shrinking problem' discussed earlier. This finding is in alignment with the work of Grau *et al.* (Grau et al., 2006), who also found that an anisotropic MRF improved robustness to various parameters.

Another key advantage of our method is the relative speed with which we are able to produce segmentations, in comparison to methods such as that of Narayanaswamy *et al.* . As the desired application of this method is to segment longitudinal data sets of large image volumes, each with upwards of billions of voxels, it is imperative that we are able to process them efficiently.

A potential area for future improvement is the limitations imposed by our edge potentials in the MRF which are tuned rather than learned. Although methods exist that are able to learn fully connected pairwise CRFs (Krähenbühl and Koltun, 2012; Lucchi et al., 2012), they require densely annotated training sets to capture the full extent of spatial interactions. The expectation of the existence of fully annotated training sets for many applications is unrealistic. One area for future work would be to focus on the suitability of semi-supervised methods to achieve fully supervised levels of performance on sparse annotations. It is possible that this may be done in

the current framework using label-transduction methods (Criminisi et al., 2011). Interesting work in the transduction and interactive learning for sparsely labelled superpixel microscopy images has also been undertaken by Su *et al.* (Su et al., 2016). A method that can take sparse image annotations and use them to leverage information from large set of unlabelled parts of the image to create high quality segmentations would be an extremely powerful tool. This would have very broad applications in novel imaging experiments where large training sets are not readily available and where there is a high time-cost in producing such a training set.

In the next chapter we will explore how more recent advances in Deep Learning can be applied to this same task. In addition to exploring the segmentation task we will focus on how we can use these methods to address situations where a pixel-labelling segmentation alone is not sufficient to extract the correct vascular network.

4

Learning Vesselness with Convolutional Neural Networks

Contents

4.1 Method Outline 92
4.1.1 Architecture 93
4.1.2 Experiments 94
4.2 Vesselness 99
4.2.1 Vesselness as a Neural Network 104
4.3 Model Combination 106
4.4 Results 108
4.5 Discussion 113

4.1 Method Outline

In this chapter we will evaluate the use of 2D CNNs for the segmentation of vasculature, as well as for a more targeted extraction of the centreline of the vasculature directly. By doing this we hope to improve our ability to delineate tightly packed and complex vasculature.

The current state-of-the-art in semantic image segmentation are the methods utilizing advances in deep learning, specifically deep convolutional neural networks (CNNs). The methods most applicable to image segmentation are the subclass of architectures known as 'fully convolutional', which produce an image of equal size to their input, after transformation to some output domain (Long et al., 2015; Chen et al., 2016c; Ronneberger et al., 2015; Zheng et al., 2015). These methods allow us to reuse many calculations by applying the same spatial priors that we apply to feature extraction, to the labelling. This results in a total number of calculations significantly lower than the previous state-of-the-art which consists of applying deep categorical classifiers in a sliding window approach. In addition to this, fully convolutional methods overcome issues of fine-grained detail by incorporating 'skip connections' to reintroduce fine detail features back into consideration after transformation into the highly abstract latent representations commonly used in deep learning, where spatial detail can often be lost. A discussion of the role of skip connections in deep learning for medical imaging has been produced by Drozdzal *et al.* (Drozdzal et al., 2016).

In this section, we will begin by evaluating the use of a state-of-the-art fully convolutional neural network for 2D segmentation on this data. We will investigate the impact of training set size and architecture size as well as the choices of loss functions. We will then consider a vessel targeted approach where, by transforming our segmentation masks, we simulate a 3D vesselness signal. Finally we will propose a hybrid model that combines both vesselness and segmentation networks. We will evaluate all proposed methods quantitatively as well as by comparing against our method from the previous chapter.

4.1.1 Architecture

The architecture we use for these experiments is the U-Net, proposed by Ronneberger *et al.*, which has enjoyed great success as a fully convolutional neural network in many different applications (Ronneberger et al., 2015). The architecture consists of a number of convolutional units, followed by sequential down-sampling. After reaching a maximum 'depth', the convolutional blocks are followed by sequential up-sampling instead. Features from the corresponding scale on the 'encoding' side are then concatenated to the features on the 'decoding' side. This means that extremely abstract representations may be learnt at a number of scales, but fine-detailed information can be re-introduced before producing a full resolution output. Although the number of downsampling levels may be varied, we heuristically choose to employ four downsampling layers. This produces a receptive field greater than 200μm, which should be sufficient for even the largest vessels present. A schematic for this architecture (Ronneberger et al., 2015) is shown in Figure 2.11. The final output of this architecture is passed through a sigmoidal activation:

$$\sigma(t) = \frac{1}{1 + e^{-t}}. \tag{4.1}$$

All training is performed using the Adam optimiser (Kingma and Ba, 2015), with parameters as suggested by the authors and an initial learning rate of 10^{-4}. A summary of the Adam optimiser is provided in Section 2.5.3. As suggested by Ronneberger *et al.* (Ronneberger et al., 2015), using a gradient descent algorithm with a strong momentum component may help to compensate for small batch sizes when training on large images. This is because each update step will contain contributions from a number of previous batches, approximating a single larger batch. This is then trained against a binary cross-entropy (BCE) loss:

$$\mathcal{L}_{bce} = \sum_i t_i \log(y_i) + (1 - t_i) \log(1 - y_i). \tag{4.2}$$

The advantages of combining a sigmoidal activation with a BCE loss function are outlined in Section 2.5.2 (Equation 2.40). As observed by many others we find an improved resistance to over-fitting and the associated artefacts by training using

Dropout at a rate of 0.5 on the final layer (Srivastava et al., 2014). A rate of 0.5 can be shown to create the largest set of 'sub-networks' by sub-sampling our network as is done with Dropout. Further discussion regarding the choice of loss functions can be found in Section 4.1.2.

4.1.2 Experiments

For the following experiments we will use the segmentation targets as described in Section 2.1.

Training Data Size Evaluation

A key topic in the application of techniques in Deep Learning to medical imaging is the dependence on large datasets. Here we will evaluate the trade-off between training set size and model capacity. In all experiments we use a dataset no bigger than could be reasonably produced by a single experimenter for a bespoke experimental set-up. Here we compare training for training sets containing 50, 100, 150, 200, 250 and 300 image tiles, each of size $512 \times 512 \times 2$. We train against a BCE loss and report values on a validation set of size 28 image tiles. For the small network we choose 32 filters at each level of depth in the network, giving 194,433 trainable weights. For the large network we choose 32, 64, 128, 256, 512, 256, 128, 64, 32 for each level of the network respectively, giving 7,846,081 trainable weights.

Training Set Size	Validation Loss - Large	Validation Loss - Small
50	0.112	0.091
100	0.098	0.090
150	0.094	0.092
200	0.085	0.093
250	0.083	0.094
300	0.082	0.095

Table 4.1: Comparisons of the effect of training set size for a large and small U-Net architecture.

Unsurprisingly, we see the best result is achieved by the largest training set. However, we may still draw a pertinent observation from this experiment. We see that for the extremely small dataset size of 50 we achieve a better validation error

using the small network. However, even though we are training on an extremely modest training set, we see that very quickly we reach a sufficient training set size such that the significantly larger network outperforms the smaller network. Performance is measured in terms of generalisation to the validation set, which is constant for all experiments listed. This provides encouraging evidence that the methods of deep learning may be effectively utilised for even small training set applications.

Loss Function Evaluation

Although in this section we choose to train against a cross-entropy loss function, some authors have suggested using a Dice coefficient based loss function (Milletari et al., 2016):

$$D(\mathbf{y}, \mathbf{t}) = \frac{-2 \sum y_i t_i}{\sqrt{\sum y_i^2} + \sqrt{\sum t_i^2}}. \tag{4.3}$$

Training against the Dice coefficient may provide favourable results when also evaluating against a Dice coefficient. However, we believe that a binary evaluation function misses some of the benefits of a probabilistic evaluation function such as cross-entropy.

If we consider the functional form of the Kullbeck-Leibler divergence (KLD) for some fixed training distribution P and a learnt distribution Q we see that

$$D_{KL}(P||Q) = \sum_i P(x_i) \log(\tfrac{P(x_i)}{Q(x_i)}), \tag{4.4}$$

$$D_{KL}(P||Q) = \sum_i \left(P(x_i) \log(P(x_i)) - P(x_i) \log(Q(x_i)) \right), \tag{4.5}$$

$$D_{KL}(P||Q) = H(P, Q) - H(P). \tag{4.6}$$

The KLD is the cross-entropy between P and Q minus the entropy of distribution P, which is a constant. Therefore by minimizing the cross-entropy we are forcing our network to most closely reproduce the statistics of the training set. This means that we can loosely interpret an output value of 0.5 as *"There is a probability of* 0.5 *that, given the input image, this pixel is part of some vasculature"*. Such a probabilistic

interpretation does not exist for a Dice based learning scheme. In fact, in practise we observe that training against Dice forces the output values to either 0 or 1.

Consider the toy example of a training set containing the values $\{0, 0, 1\}$ in a binary classification problem. If we consider the optimal result for a single prediction under the two different loss functions we see the following:

$$\ell_{Dice} = \frac{-2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} v \\ v \\ v \end{pmatrix}}{1 + v\sqrt{3}}, \tag{4.7}$$

$$\ell_{Dice} = \frac{-2}{\frac{1}{v} + \sqrt{3}}. \tag{4.8}$$

For v in the range $[0, 1]$, ℓ_{Dice} is clearly minimised by $v = 1$. In the case of cross-entropy we see:

$$\ell_{bce} = -\log(1 - v) - \log(1 - v) - \log(v), \tag{4.9}$$

$$\frac{\partial \ell_{bce}}{\partial v} = \frac{1}{1 - v} + \frac{1}{1 - v} - \frac{1}{v} = 0, \tag{4.10}$$

$$v = \frac{1}{3}. \tag{4.11}$$

In this instance, the cross-entropy loss gives us some quantitative insight into the uncertainty present in the prediction, whereas the Dice based loss tends to reduce false negatives at the cost of increasing false positives. In this case the Dice measure rewards us for being over-zealous in the face of uncertainty, which may be undesirable for quantitative evaluations beyond simply maximising a Dice overlap.

We will explore this further with some experiments on real data. We train one network to minimise a cross-entropy loss while reporting the corresponding Dice coefficient loss for each epoch and another network to minimise the Dice coefficient loss while also reporting the corresponding cross-entropy loss for each epoch.

By considering the training loss histories in Figure 4.1, we can make a number of observations. Firstly, we see that the Dice coefficient based loss (top left) appears more resistant to over-fitting. This is could be inferred in the fact that the validation Dice error (green) does not rise after reaching some minima. However, we see this is also the case when training using the BCE loss (top right, green). Even though

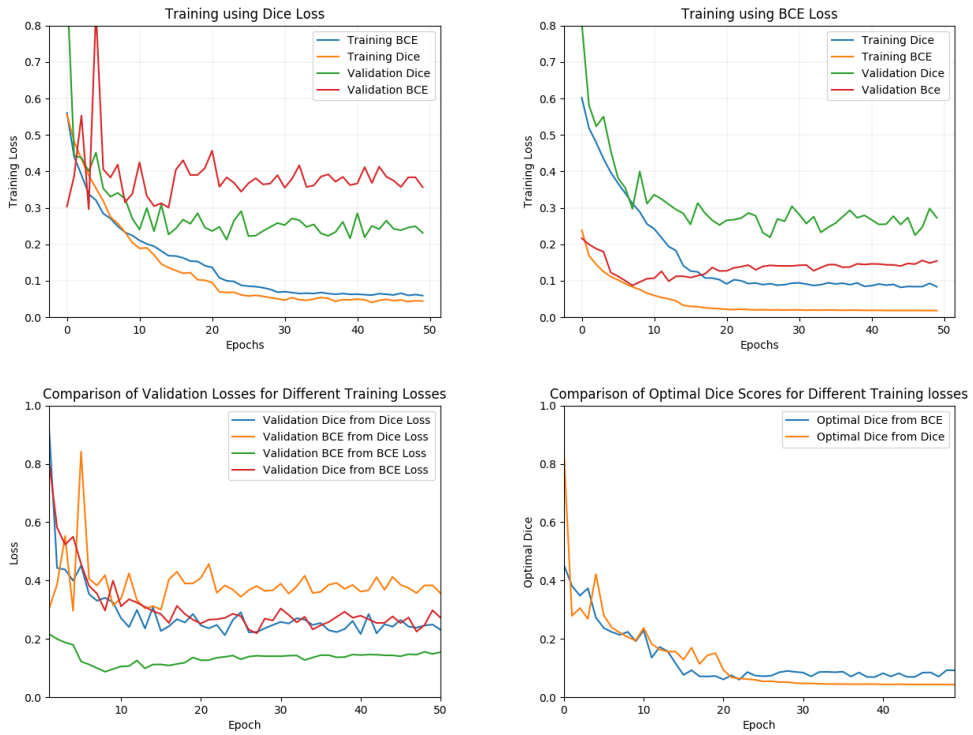


Figure 4.1: Top Left: Network trained to optimise a Dice based loss, corresponding BCE values are reported for each epoch. Top Right: Network trained to optimise a BCE based loss, corresponding Dice values are reported for each epoch. Bottom Left: Comparison of validation losses from both networks. Bottom Right: Comparison of Dice score at an optimal threshold for both networks.

over-fitting is occurring after approximately epoch 10 (red) this is not reflected in the validation Dice score (green), which remains constant. This suggests that early stopping would not be an effective regularisation technique when training using a Dice coefficient based loss and that an optimal Dice based fit corresponds to an over-fitting in terms of a cross-entropy based loss. This makes some degree of intuitive sense as a probabilistic comparison between training sets naturally contains more information than a binary comparison since one can be derived from the other. When comparing the validation losses for both methods (bottom left), we observe that the validation Dice score is comparable for both methods, however, the validation BCE score is superior in the case of training with a BCE loss.

As we have previously seen, the Dice coefficient naturally tends towards binary solutions rather than fuzzy probabilistic ones, as in the case of cross-entropy. It is therefore unfair to compare the two methods directly. Instead we should perform

a binarisation step to each method before comparison, as would be the case in a true segmentation evaluation. In order to achieve this we compute the optimal threshold, T_{opt}, (over the entire training set) and apply this threshold to the network outputs at this value such that

$$\bar{y}_{ij} = \begin{cases} 1 & \text{if } y_{ij} \geq T_{opt} \\ 0 & \text{otherwise.} \end{cases} . \tag{4.12}$$

We can then compare an optimally thesholded Dice score between the two networks, which have been trained using different loss functions. In these results (Figure 4.1, bottom right) we see that both methods achieve comparable optimal Dice scores. We note that the optimal value corresponds roughly with the point at which the validation Dice score stops decreasing (top right, green). We therefore suggest training using the BCE loss and performing an early stopping on the validation BCE score to preserve probabilistic information and early stopping procedure on a validation Dice score in order to optimise Dice overlap scores.

In addition to an optimally binarised comparison, we also test robustness to this choice of binarisation threshold, by means of a Receiver Operating Characteristic (ROC) analysis, where we sweep over all possible thresholds, plotting the true-positive rate against the false-positive rate. From this we see that although both methods score very well on this analysis, the BCE method performs best, with an Area Under Curve (AUC) of 0.984 as shown in Table 4.2.

Loss function	AUC
Dice Coefficient	0.954
Cross-Entropy	0.984

Table 4.2: Comparison of AUC from ROC analysis of Dice and cross-entropy loss functions.

For the reasons given in this section we will proceed with cross-entropy based loss functions for the remainder of this thesis. We also note that with both loss functions, we were able to achieve an optimal Dice score of approximately 0.95, a significant improvement on all methods from the previous chapter in terms of segmentation overlap.

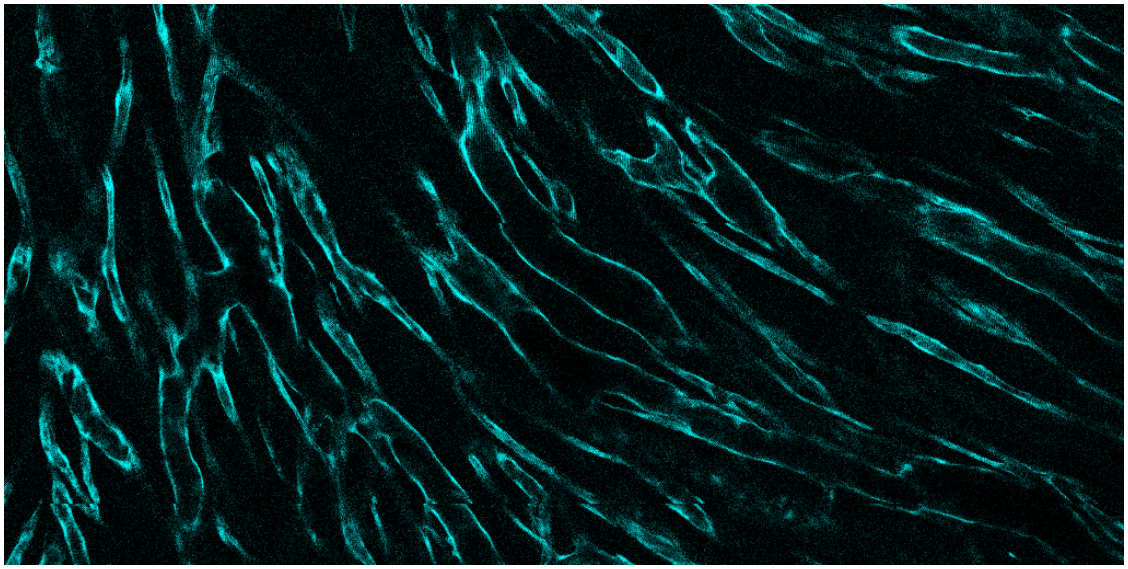


Figure 4.2: Tightly packed vasculature, endothelium channel.

4.2 Vesselness

Although we are able to achieve very high Dice overlap on these images (approximately 0.95), the segmentation of vasculature is unique to most other segmentation tasks in that we are interested jointly in some combination of geometry and topology. In the case of highly packed vasculature, as is often the case in tumours (see Figure 4.2), we may achieve a segmentation with a very high Dice overlap score, but which entirely misses the topological structure of the vasculature. This leads us to believe that a simple pixel-labelling of the vasculature is not sufficient for this task.

In order to perform analysis on the shape of the vasculature, we wish to extract a medial representation from the segmentation, a skeleton. This skeleton should preserve the topology of the vascular network, allowing us to perform measurements. The standard and by far the most efficient way for us to do this is to reduce the segmentation using an iterative thinning approach until we are left with a simply connected skeleton. However, we encounter issues if two sections of parallel vasculature happen to touch, or come close, as our thinning algorithm will infer that they are topologically linked, when this is not the case. In Figure 4.3 we see a demonstration of this phenomenon. Here, the segmentation is shown in white and the derived centreline is shown in red. In the ground truth case (top) we

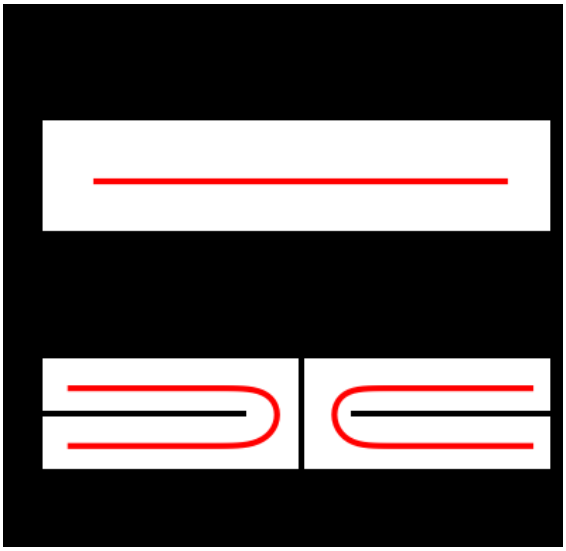


Figure 4.3: Small errors in segmentations may cause very large changes in derived centerlines. Segmentation shown in white and predicted centreline shown in red.

see that the segmentation takes the form of a simple rectangle, and the derived centreline (acquired by thinning) is a straight line running longitudinally, as expected. In the pathological case (bottom) we have removed a small number of pixels, altering the topology and fundamentally changing the derived centreline. These two segmentations share a Dice overlap of approximately 0.95 yet the centrelines derived from the pathological case are incorrect.

The classical way in which vascular segmentation is performed is to use some model of vesselness, a quantity which is high along the centre of a section of vasculature and low elsewhere. Typically this is based upon local curvature relationships (Frangi et al., 1998; Sato et al., 1997; Lorenz et al., 1997; Krissian et al., 1998), however, the irregular and non-tubular morphology of our vasculature will not permit such a simple approach. Instead we will attempt to regress this vesselness quantity directly from our images using another CNN.

In order to do this we transform our training data to provide some proxy towards vesselness/centrelines. Let $I(x)$ be image intensity at point x and let $D(x)$ be defined as follows:

$$D(x) = \begin{cases} \text{dist}(x, 1 - I), & \text{if } I(x) = 1 \\ 0, & otherwise \end{cases} \tag{4.13}$$

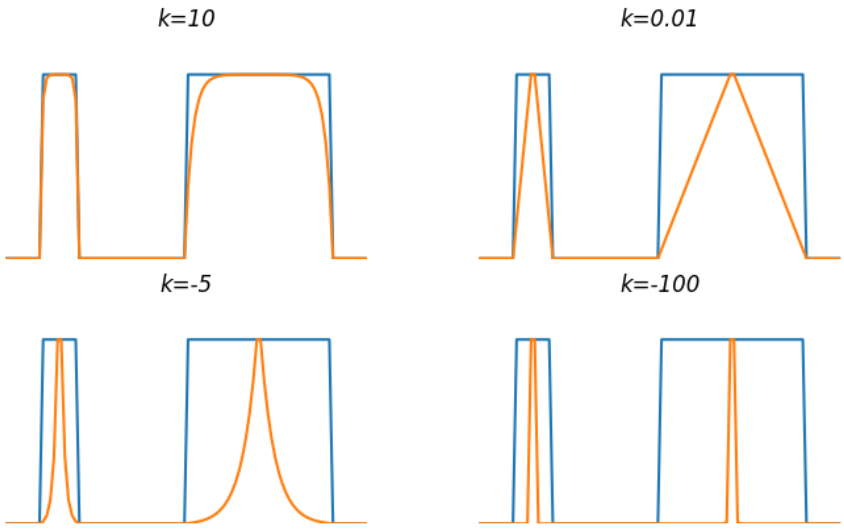


Figure 4.4: The original segmentation is shown in blue, and the resultant vesselness transform according to Equation 4.14 is shown in orange. The vesselness transform is visualised for a range of values of k. Adjusting this changes the specificity of the skeleton we would like to learn.

where the $\text{dist}(x, Y)$ gives the shortest distance from the point x to a non-zero entry in Y.

We then transform the field D as follows:

$$V_k(x) = \frac{e^{-k\chi(x)} - 1}{e^{-k} - 1} \tag{4.14}$$

$$\chi(x) = \frac{D(x)}{M(x) + \delta} \tag{4.15}$$

where $M(x)$ is the local maximal value in a sliding window of size (x_M, y_M, z_M) about the point x in the distance map. This transform normalises the signal such that now we have a value of 1 along the vessel centreline, decaying to zero away from these regions.

This quantity can be tweaked by adjusting the value of k. The limit $\lim_{k\to\infty} V_k$ gives the original segmentation whereas $\lim_{k\to-\infty} V_k$ returns the centreline only. By adjusting k we can achieve a continuum of intermediate representations. We found $k = -5$ to be an effective compromise. Although the skeleton is what we would like to learn, adding tails to this distribution about the skeleton provides



Figure 4.5: Left: Slice through endothelium channel, Centre: manually annotated training data, Right: training data transformed according to Equation 4.14 with $k = 3$.

additional context for the learning algorithm and should help create gradients to the optimal solution. Such a transformation leaves us with a signal that takes the value of 1 along a vessel 'centreline' and decreases to zero elsewhere. A key point here is that information about the location of the centreline is shared among neighbouring slices, allowing a 2D network to learn some 3D cues. As in Section 2.1 we now have an additional volume corresponding to each manual ground truth segmentation which we will refer to as the *vesselness* target.

In Figure 4.5 we see the appearance of the transformed training data. For vesselness training we exchange the sigmoid final layer used for the segmentation with a ReLU activation:

$$ReLU(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{otherwise} \end{cases} \tag{4.16}$$

We then optimise a mean squared error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_i (g(x_i) - V_k(x_i))^2 \tag{4.17}$$

where $g(x)$ is the network output at location x and V_k is the derived centreline value given in Equation 4.14.

In Figure 4.6 we see the output of our vesselness trained network. As desired we are highlighting along the centre-line of the vasculature, helping us not only detect the vasculature but to extract the topology of complex regions such as the loop visible in the top left corner of the image. Our initial motivation in proposing

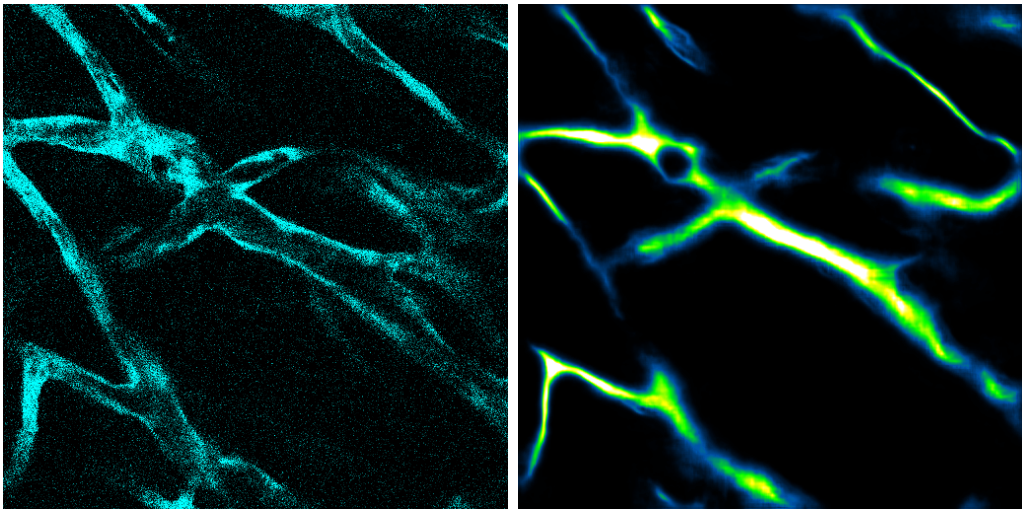


Figure 4.6: Image filtered to produce learned vesselness map.

this approach was due to the fact that our data was not amenable to standard vesselness filtering due to the non-standard appearance and non-tubular geometry. However, it is worth noting that the output now produced shares an appearance with vasculature traditionally seen in medical images for both vasculature and neuronal imaging, transforming our non-standard image data into a domain in which it may be amenable to more standard vessel extraction techniques.

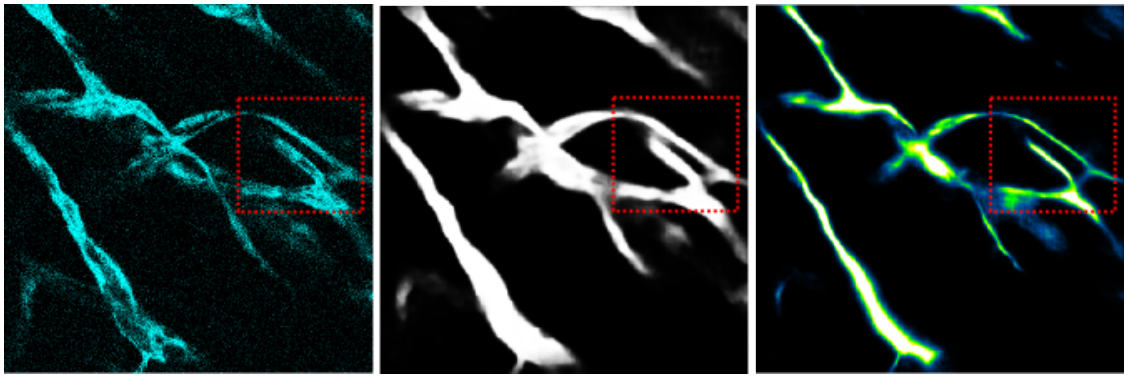


Figure 4.7: Left: Endothelial channel, Center: Segmentation network, Right: Vesselness network.

In Figure 4.7 we see the advantage of our vesselness filtering. In the region within the red square we see two parallel vessels. Although the segmentation network has done a good job of identifying all of the vasculature in the image, it has incorrectly connected these two vessels. In the vesselness image we see that they are clearly distinguished.

4.2.1 Vesselness as a Neural Network

As an aside, we will show that the implementation of a vesselness algorithm is trivially possible using neural network machinery. It therefore should come as no surprise that we are able to learn a similar model via gradient descent.

Vesselness is a classical measure for vessel segmentation that models a tubular structure by the eigenvalues of the local Hessian matrix. Traditionally this has been used to segment or enhance vessel structures that already conform to a tubular shape, however, some images of vasculature do not conform to this simple model.

Frangi Filter

The Frangi filter (Frangi et al., 1998) works on the consideration that the appearance of vasculature in an image causes a ridge in intensity space. We are able to quantify ridges by considering the local Hessian matrix which gives information on curvature.

$$\mathcal{H}_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} \tag{4.18}$$

The eigenvectors and eigenvalues of the Hessian matrix provide the direction and magnitudes of the principle curvatures at this point of the surface.

Letting $\{\lambda_1, \lambda_2\}$ be the eigenvalues, such that $|\lambda_1| > |\lambda_2|$ then,

$$S = \sqrt{\lambda_1^2 + \lambda_2^2} \tag{4.19}$$

$$R_B = \frac{|\lambda_2|}{|\lambda_1|} \tag{4.20}$$

$$V = \begin{cases} 0 & \lambda_2 < 0 \\ \exp\left(R_B^2\right)(1 - \exp\left(S^2\right)) & \text{otherwise} \end{cases} \tag{4.21}$$

CNN implementation

All parts of this can be implemented using fully differentiable network operations. We can implement the computation of the Hessian elements using two Convolutional layers. The first later is hard coded with 2 orthogonal edge detection filters, e.g. Sobel, forming a weight tensor of size $[2, 1, D, D]$. The second layer is implemented with duplicates of the edge detection filters. These filters are alternately zero padded in the 2nd dimension to force them to operate on the correct 1st derivative to form

the mixed derivatives. This produces a weight vector of size $[4, 2, D, D]$. After these two convolutional layers we are left with a tensor of size $[1, 4, N, N]$ where the feature images in the 2nd dimension represent the 4 mixed derivatives of the Hessian matrix.

For the eigenvalues of a 2×2 matrix there is a simple closed form solution

$$Det(A - \lambda I) = 0 \tag{4.22}$$

$$Det\left(\begin{bmatrix}(a-\lambda) & b \\ c & (d-\lambda)\end{bmatrix}\right) = 0 \tag{4.23}$$

$$(a - \lambda)(d - \lambda) - cb = 0 \tag{4.24}$$

$$\lambda^2 - (a + d)\lambda + (ad - cb) = 0 \tag{4.25}$$

$$\lambda^2 - Tr(A)\lambda + Det(A) = 0 \tag{4.26}$$

$$\lambda = \tfrac{Tr(A)}{4} \pm \sqrt{\tfrac{Tr(A)^2}{4} - Det(A)} \tag{4.27}$$

This can be implemented by splitting and merging the various channels comprising the Hessian matrix values. We can rearrange the eigenvalues into λ_{max} and λ_{min}, such that

$$\lambda_{max} = \begin{cases} \lambda_1 & \text{if } |\lambda_1| > |\lambda_2| \\ \lambda_2 & \text{otherwise} \end{cases}, \tag{4.28}$$

and vice versa.

The gradient is also defined as

$$\frac{\partial \lambda_{max}}{\partial \lambda_i} = \begin{cases} 1 & \text{if } \lambda_{max} = \lambda_i \\ 0 & \text{otherwise} \end{cases}, \tag{4.29}$$

and similarly for λ_{min}.

The vesselness criterion V again can be implemented using simple transformations and multiplication layers, additionally we approximate the requirement that $\lambda_{max} < 0$ using a sigmoidal activation.

$$V = \exp\left(-\tfrac{|\lambda_{max}|}{|\lambda_{min}|}\right)\left(1 - \exp\left(-(\lambda_{max}^2 + \lambda_{min}^2)\right)\right)\sigma(-\lambda_{max}) \tag{4.30}$$

This vesselness layer can be applied at multiple scales by passing input data through a number of smoothing convolutions, with a copy of the data sent through

a vesselness unit at each scale. The total stack of vesselness outputs can then be concatenated along the scales axis either by taking the maximum response or by some learned combination.

We have posed this as hypothetical to demonstrate the feasibility of learning vesselness representations. However, this suggests a natural way in which we could use this to 'invert' the role of the vesselness measure and use it as an unsupervised way to assess the 'vessel' quality of learned representations in from CNNs. A semi-supervised learning scheme could be derived from a 'vesselness' unit such as this that aims to maximise vesselness as an auxiliary loss. In order to implement this we have suggested a fully differentiable vesselness unit which can be inserted into any neural network architecture. In addition to this we suggest a multi-scale vesselness unit to be used to apply this at multiple scales as in the traditional vesselness algorithm. Although we present the implementation of the standard Frangi Vesselness algorithm, more complex eigenvalue algorithms could easily be implemented.

4.3 Model Combination

Previously, we proposed training our model to reproduce some vesselness signal. However, this vesselness filtering comes at the cost of information regarding the shape and size of the vasculature. In a classical vesselness filtering setting this could be approximated by analysing the scale space at which the maximum response was achieved. In addition to this we find that the vesselness model is more prone to artefactual responses than the segmentation model.

In order to address all of these issues we propose to combine the output of both the segmentation and vesselness networks and show that the resultant segmentation contains a full segmentation of the vasculature that also preserves the shape of the vasculature, clearly distinguishes between vessels and is not prone to artefacts. It is known that machine learning algorithms trained and applied as an ensemble almost always outperform those acting alone, a fact that is made apparent by examining the leader boards for any of the public machine learning challenges. This effect is exploited and approximated by the use of Dropout during training (Srivastava et al.,

2014). However, another kind of ensemble is the generalist/specialist ensemble, suggested by Hinton *et al.* (Hinton et al., 2015), where we have a generalist network which focuses on the broad strokes and a specialist network which deals with fine details. It is this combination of networks that we are exploiting here. By this interpretation, the segmentation network takes the role of the generalist, loosely identifying the regions which are vasculature and the vesselness network takes the role of the specialist, identifying the centreline within the vasculature.

In their paper proposing the 'distillation' of knowledge from ensembles of networks, Hinton *et al.* (Hinton et al., 2015) propose the following as a solution for combining the outputs from an ensemble.

$$q_m = \arg\max_q \left(D_{\mathrm{KL}}(p_g||q) + D_{\mathrm{KL}}(p_s||q)\right) \tag{4.31}$$

where

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i)\, \log \frac{P(i)}{Q(i)}. \tag{4.32}$$

In fact, in the case of a binary output where $P(2) = 1 - P(1)$, we can show that this optimal solution is given by the arithmetic mean of the distributions p_g and p_s:

Let $p_g = [p_g, 1 - p_g]$ and $p_s = [p_s, 1 - p_s]$ and $q = [q, 1 - q]$

$$L = p_g \log\left(\tfrac{p_g}{q}\right) + (1 - p_g) \log\left(\tfrac{1-p_g}{1-q}\right) + p_s \log\left(\tfrac{p_s}{q}\right) + (1 - p_s) \log\left(\tfrac{1-p_s}{1-q}\right) \tag{4.33}$$

$$\frac{\partial L}{\partial q} = \tfrac{p_g}{q} - \tfrac{1-p_g}{1-q} + \tfrac{p_s}{q} - \tfrac{1-p_s}{1-q} = 0 \tag{4.34}$$

$$(1 - q)p_g - q(1 - p_g) + (1 - q)p_s - q(1 - p_s) = 0 \tag{4.35}$$

$$q = \tfrac{1}{2}(p_s + p_g) \tag{4.36}$$

This suggests that taking the arithmetic mean of our outputs would also be a justified approach from the interpretation of the segmentation and vesselness networks as a generalist and specialist ensemble.

A visual inspection of the two methods applied on whole tumours in Figure 4.9 appears to show improved definition to the segmentation, as seen in Figure 4.8. By considering a sagittal view as shown in Figure 4.10 we see that this

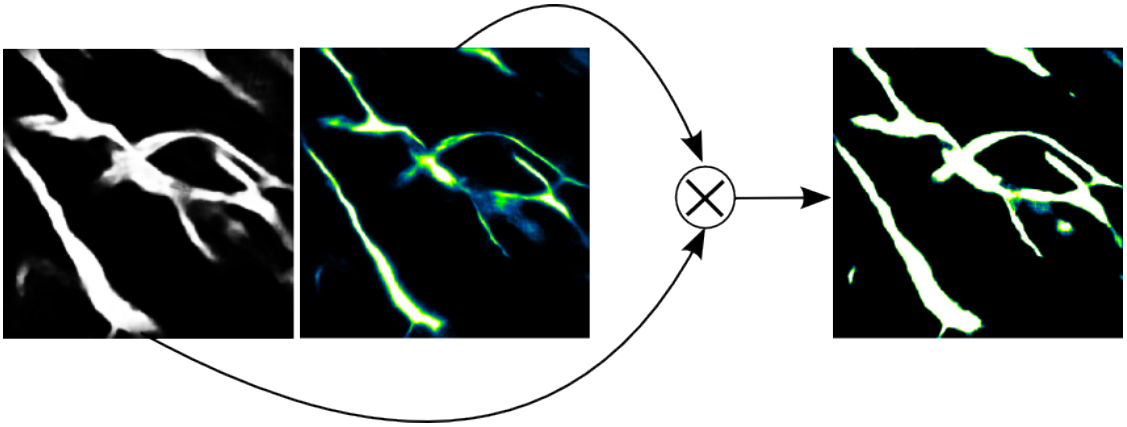


Figure 4.8: Segmentation image multiplied by Vesselness image.

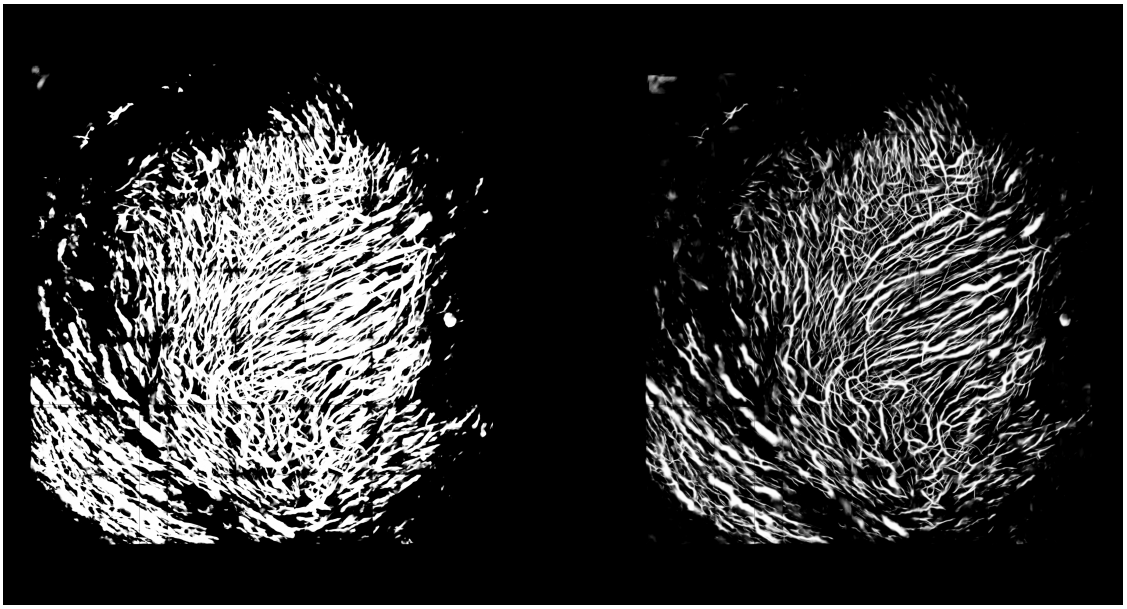


Figure 4.9: Left: Segmentation network, Right: Vesselness network.

improved definition also helps to separate structures in 3D as desired. The vesselness signal is more tightly localised to the center of the vessel improving our ability to resolve tightly packed but separate structures, seen markedly towards the left hand side of Figure 4.10.

4.4 Results

Here we present results for experiments using the segmentation network, the vesselness network and the two combination networks motivated in Section 4.3.

In the previous chapter we assessed the method using a combination of the Dice

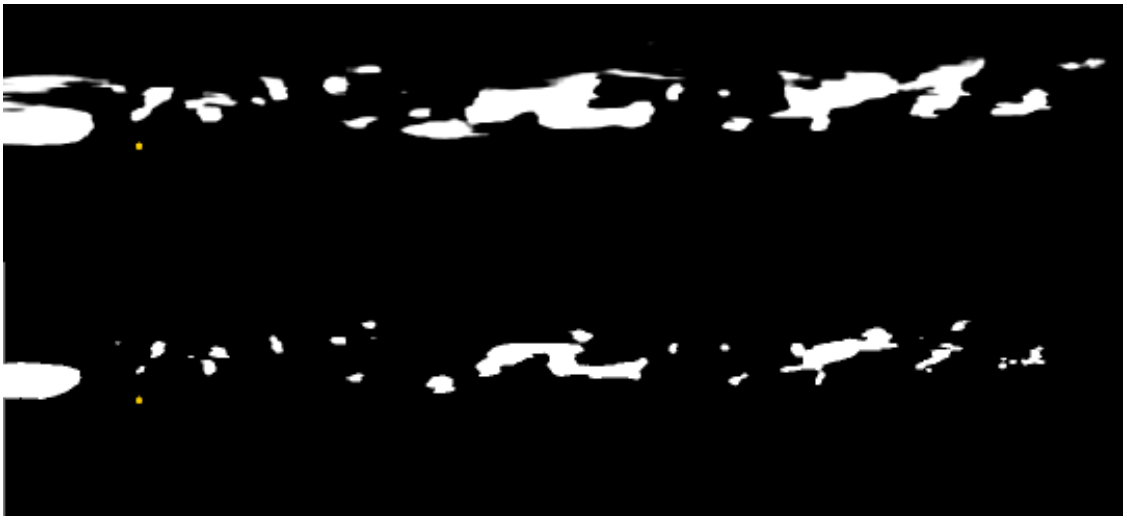


Figure 4.10: Top: Segmentation network, Bottom: Vesselness network.

coefficient and the Hausdorff average distance from the predicted skeleton to ground truth, however, this may not sufficiently penalise under-segmentation of the skeleton. For the following assessments we consider the symmetric Hausdorff average:

$$d_{sym}(A, B) = \max(d_H(A, B), d_H(B, A)). \tag{4.37}$$

where d_H is defined as in Equation 3.15. Dubuisson suggested taking either the maximum value or the average value to form a symmetric measurement. Here we prefer the maximum, as it prevents a good score in one direction from compensating for a poor score in the reverse direction. Before computation of this distance we perform a thresholding and thinning operation (Pudney, 1998). Although many more sophisticated skeletonisation algorithms exist, we would like to emphasise our aim of developing a network for which the 'skeletonisation' step is a trivial one and has been entirely learnt by the network. We report the best score achieved over all possible thresholds.

For these experiments we test on 6 unseen image volumes and we train on the 8 image volumes used in the previous chapter. The test data used here is also considerably more challenging than that used in the previous chapter due to more tightly packed and tangled vasculature. These cases come from Group 3 as described in Section 2.1.1, with the skeleton targets being used to provide a ground truth.

For comparison with the previous chapter we include the results of the Supervoxel MRF (SV-MRF) method as well. The competing methods are as follows:

1. **Segmentation network (*seg*)**: U-Net network as described in Section 4.1.1, trained against the segmentation target volumes.

2. **Skeleton network (*skel*)**: U-Net network as described in Section 4.1.1, trained against the vesselness target volumes.

3. **Sum network (*sum*)**: The arithmetic mean of the outputs of *seg* and *skel*, as motivated in Section 4.3.

4. **Sum network (*mult*)**: The geometric mean of the outputs of *seg* and *skel*, as seen in Figure 4.8.

5. **Supervoxel MRF (*sv mrf*)**: For comparison we also include the proposed method from Chapter 3.

The results of this analysis are shown in Figure 4.11. Using a one-tailed Wilcoxon's Signed-Rank test ($\alpha = 0.05$), and correcting for multiple comparisons using Bonferroni-Holm correction, we are able to show a significant improvement using the *mult* method over both the original *seg* method and the *sv mrf* method. We also note that in addition to the significance observed in the quantitative measurements, our proposed method most consistently produces segmentations that, on visual inspection, most resemble the ground truth. Errors of 20 or greater in Table 4.11 tend to reflect either a severe over segmentation or a severe under segmentation and would not be considered an accurate representation of the network. In the case of the *sv mrf* and *seg* methods we find the most common failure mode on this test set to be an over-segmentation and tendency to 'merge' together large regions of tightly packed vasculature. In the case of the *skel* method we find that the most common failure mode is under-segmentation.

For a further analysis, we compare the results achieved on these extremely difficult cases (from Group 3) to the analysis performed in the previous chapter

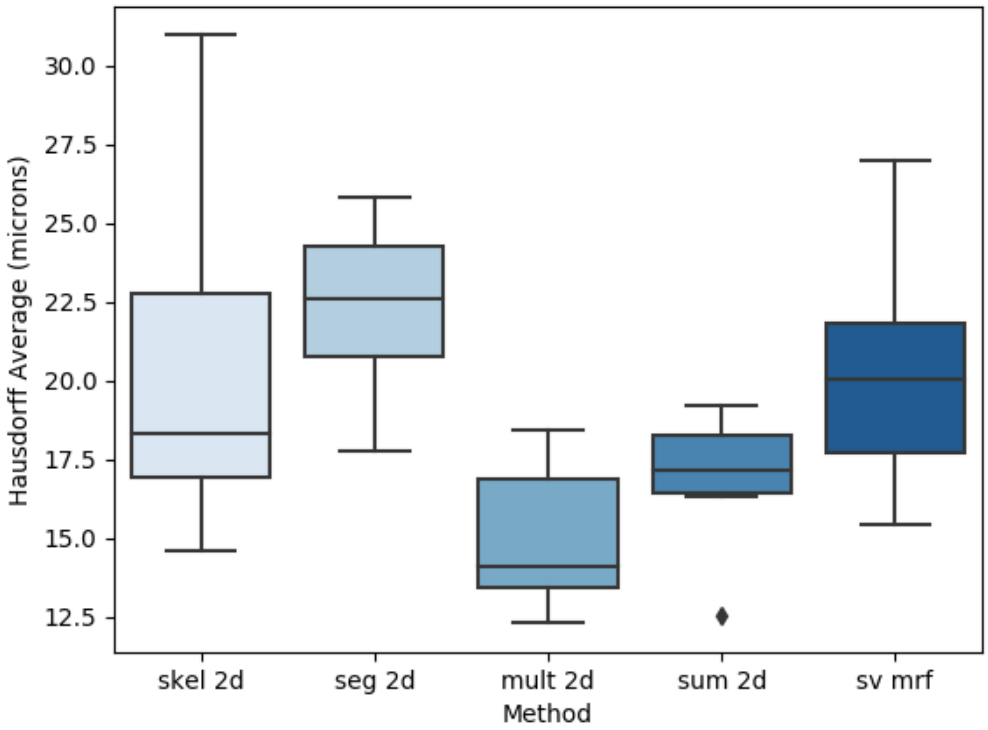


Figure 4.11: Box plots showing the comparison of the various methods under the Hausdorff average (max) measurement.

on the simpler cases (from Group 1). The results of this are shown in Figure 4.12. Here we see that, while the combination networks remain superior in the combined analysis, the vast majority of the improvement comes from their ability to segment the complex Group 3 cases. We also see that the methods proposed here consistently out perform the *sv mrf* method, even on the expanded data set.

Some authors have highlighted the fact that a Hausdorff measure may not sufficiently penalise errors in topology (Mayerich et al., 2012). Therefore, in addition to this Hausdorff analysis we consider a topological measure of agreement based on the graphical structure of the network. For this we consider metrics from the NetMets framework proposed by Mayerich *et al.* (Mayerich et al., 2012). Although other methods exist such as the Tree Edit Distance (Heumann and Wittum, 2009), Path2Path (Basu et al., 2011), Elastic Shape Analysis (ESA) (Mottini et al., 2015) and the DIADEM metric (Gillette et al., 2011), we prefer the NetMets approach as

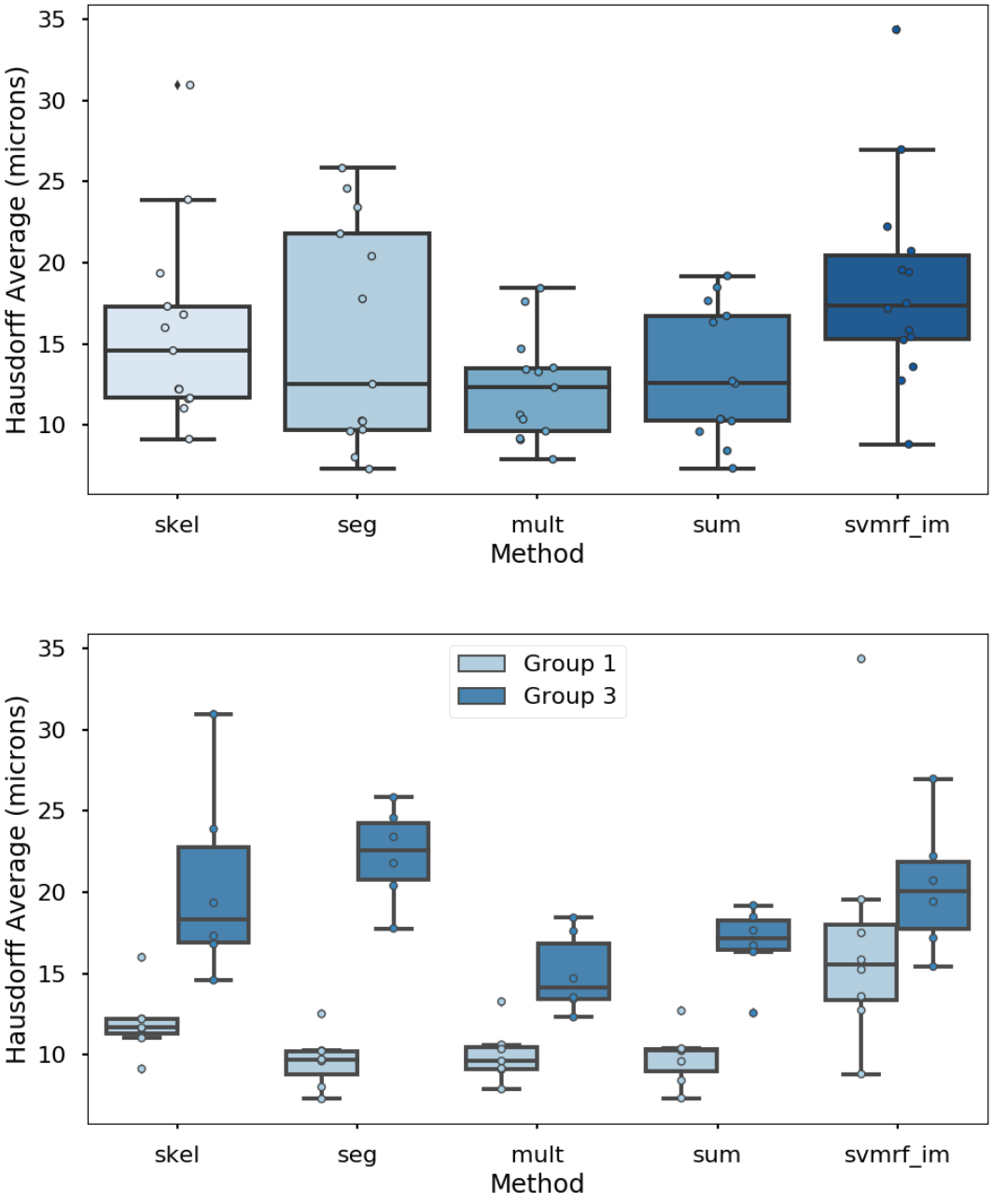


Figure 4.12: Comparison of Hausdorff Analysis including examples from Group 1 and Group 3 (top). Comparison between Hausdorff results from Group 1 and from Group 3.

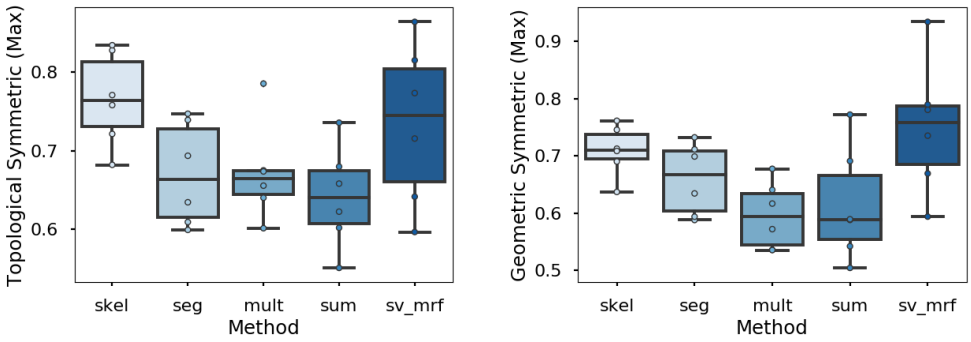


Figure 4.13: Comparison of topology scores T_{max} and G_{max} for all methods.

we are not constrained to tree-like or 'rooted' topologies, such as those commonly found in neuron tracking tasks. One caveat that should be noted in the case of the NetMets approach is that we are only measuring local rather than global topological agreement. However, we believe that this is desirable as our vascular structures do not exhibit such a top-down, hierarchical structure. A full explanation of this method may be found in Appendix A. Here we consider the T_{max} and G_{max} measurements. The results of this are shown in Figure 4, here we report values for an optimally chosen threshold for each case. Here we see that the methods presented in this chapter, specifically the combination methods *skel* and *sum*, clearly outperforms the previous *sv mrf* method. For comparison we also found that the method of Narayanaswamy (Narayanaswamy et al., 2010), used as a comparison in the previous chapter, scored approximately $T_{max} = 0.88$ and $G_{max} = 0.85$. Although we lack statistical power due to a fairly small test set, we find that that in the case of G_{max}, the *mult* method greatly outperforms the previous *sv mrf* method.

4.5 Discussion

In this chapter we have developed a method to approximate a 3D vesselness function using 2D CNNs. The network trained on this data should enhance only the center of a vessel, rather than the entire vessel as with segmentation. The aim of this is to increase the specificity of our vessel detection and thus improve the ability to correctly extract the vasculature and its topology. Although we found no significant

improvement when using the vesselness trained network alone over the segmentation trained network, we found that using the two networks in conjunction led to an improvement. When taking the geometric mean of the two methods provided a significantly improved result in terms of our chosen measures of the Hausdorff average distance (Dubuisson and Jain, 1994) and the topological measurements of the NetMets framework. We also saw favourable improvements in the case of taking the arithmetic mean of the two networks, as motivated in Section 4.3. This tells us that there is value in learning this vesselness function, as we force the network to learn separate but complementary information to the segmentation alone. By comparing the results on the complex images to the results from the relatively simpler cases from Chapter 3, we see that the inclusion of the vesselness network was vital in the case of the complex cases, but did not improve the results of the segmentation network in the simpler cases. This agrees with our intuition that the role of the vesselness network is to assist the localisation of the centreline in complex networks, where segmentation alone can cause topological errors.

Although we believe that this represents an improvement to learning segmentations only, we suggest a number of limitations that still remain with this method: firstly the method is still extremely vulnerable to slice misalignment within the test image. Since this is still an inherently 2D method, we have no power to prevent discontinuities in the input image (due to motion or otherwise) from causing discontinuities in the predicted segmentation or vessel network. Secondly we are heavily dependent to tuning an optimal threshold. Although we have seen that we are able to achieve an improved delineation of the network using this approach it is necessary to tune an optimal threshold to balance between under and over segmentation of the network. Finally, in order to encode spatial information about the 3D location of the center of the vessel we necessarily reduced the specificity of our signal by introducing tails into the function given in Equation 4.14. This trade-off in specificity requires that we apply a thresholding and an additional skeletonisation step in order to reach our derived vessel centreline.

In this chapter we have demonstrated improved results by utilising multiple networks to approach different aspects of a given task. However, there is likely a large amount of redundancy within the representation learning aspects of the two networks, especially in the earlier layers. In future work, it may be of interest to attempt to extend this work in a multi-task learning framework, using a single, multi-headed network rather than two independent networks. This may improve robustness and would reduce computational load.

In the next chapter we will work toward reducing our dependency on an optimal threshold as well as improving the specificity of our centreline detection. We will also address the limitations associated with purely 2D methods by extending to a 3D framework.

5

Learning 3D Skeletons with Convolutional-Recurrent Networks

Contents

5.1 Introduction 118
5.2 Related Work 120
5.3 Materials 121
5.4 Methods 122
5.4.1 Architectures 122
5.5 Results 129
5.5.1 Results on Synthetic Data 130
5.5.2 Results on Real Microscopy Data 133
5.6 Discussion 135

5.1 Introduction

In the previous chapter, we trained a 2D CNN to produce a 3D vesselness function. This was used as a proxy to produce a vessel centreline that we can use to accurately analyse the structure and form of our vascular networks. Although this appears to be an effective alternative to just learning the segmentation, this approach may suffer from a strong dependence on thresholding and a tendency for weak or blurred signal in regions of uncertainty and for large vessels. An alternate approach to this vesselness function would be to attempt to learn the skeleton directly, which is an extremely difficult task for a 2D network. Consider the example of a 3D centreline that happens to lie entirely on a single plane of an image volume. During training, the CNN attempting to learn from the two planes either side of this plane would have no information regarding the location of this centreline, even though there will clearly be valuable image features for prediction. In the previous case of a vesselness function, this was addressed by having our transformation extend in all directions, as in Equation 4.14. However, this came at the cost of a loss of localisation of the response. In order to learn such a sparse 3D structure, we must move to a 3D network, so as to be able to propagate this information between slices directly, rather than relying on some proxy signal.

The majority of the work in semantic image segmentation involves the application of CNNs to 2D images, largely in the domain of natural images. Although CNNs have been extended to consider 3D interactions, as has been discussed in Section 2.5.7, we believe that this may be sub-optimal for applications such as ours, where the image does not represent an isotropic sampling of the domain. Since both the number of parameters and the computational complexity will increase exponentially with the effective dimension for convolutions, we restrict ourselves to convolution only in the information-rich x-y plane. However, there is important contextual information contained in neighbouring planes that we must exploit in order to produce a satisfactory extraction of the vasculature. In order to tap into this information without overly burdening ourselves with model complexity, we will use a variant of the Long Short-Term Memory (LSTM) recurrent neural network

units (Hochreiter and Schmidhuber, 1997). The recently popularised Convolutional Long Short-Term Memory (ConvLSTM) units (Shi et al., 2015) provide a way to efficiently model sequences of images. These networks modify the LSTM model of recurrent neural networks, replacing their inner product operations on feature vectors with kernel convolution operations on images. In this work we will apply these ConvLSTM units to tie together the detection results from each plane to create a meaningful 3D structure for the vasculature present in the image volume.

The motivation for this approach comes from how humans interact with image volumes of this kind which itself is a product of the imaging modality. The vast majority of the useful information exists in the high resolution x-y plane, due to the z-stack method of imaging of imaging. When we engage with images of this kind we find it most natural to scroll up and down the z-stack rather than considering sagittal or coronal views of the stack. In this framework, the CNN represents the human visual system, analysing each individual slice, whereas the ConvLSTM unit represents the scrolling motion, combining understanding from each individual slice, rather than trying to model the slice appearance jointly. Learning a 3D CNN would mean that the kernels are required to jointly learn an appearance model and a physical motion model, which is completely decorrelated from the appearance model. The source of motion present in our images is derived from the breathing artefacts inherent in live animal imaging. It appears to be sub-optimal to attempt to model appearance and breathing motion jointly, therefore this motivates us to attempt to separate these two aspects of the vision problem.

A common pipeline for the extraction of vasculature is segmentation followed by a medial axis extraction, which is not always possible. In images containing tightly packed vascular structures it may be the case that a segmentation will be unable to separate extremely close, or even touching, sections of vasculature. If we then attempt to extract the vascular topology from this segmentation, it may not represent the true network structure of the underlying vasculature. In order to address this, in this chapter we propose to learn to extract the medial representation of the vasculature directly, by training our network to reproduce skeletons of the

vasculature, derived from manual segmentations, rather than training against the segmentations themselves or some proxy vesselness function. The key difficulty arises from the fact that the medial representation of a 3D structure inherently requires a 3D view of the domain and that, unlike segmentations, it is extremely sparse. This cannot be achieved by applying a CNN to each slice in our image. Instead, we propose to link together a shared CNN network applied to each image slice using ConvLSTM layers. This provides a mechanism for the sparse information to be shared between the feature extraction CNN networks on neighbouring slices, even when no information about the skeleton may be present directly for that slice of the image. In addition to this we will analyse the role played by different loss functions and also the importance of applying our recurrent layers in a bidirectional sense, as has been explored in other applications such as video analysis (Graves, 2013).

5.2 Related Work

A number of authors have attempted to apply Deep Learning methods to segmentation of vasculature. However, the vast majority of effort in this area is applied to segmentation of vasculature in optical imaging of the retinal fundus (Liskowski and Krawiec, 2016; Fang et al., 2015; Li et al., 2016). Prentašic *et al.* have applied deep learning methods to coherence microscopy images of micro-vasculature (Prentašic et al., 2016) but they consider only 2D images in which the vasculature already has an enhanced appearance. Teikari *et al.* have applied CNNs to the task of segmentation in 3D fluorescence microscopy images of micro-vasculature, similar to ours (Teikari et al., 2016). However, they consider contrast enhanced vasculature, and only concern themselves with the segmentation of the vasculature, rather than extracting a medial representation. 3D CNNs have been applied successfully to brain lesion segmentation by Kamnitsas *et al.* (Kamnitsas et al., 2016b) with their Deep Medic architecture. 3D CNNs have also been explored in the V-Net architecture of Milletari *et al.* (Milletari et al., 2016) for prostate segmentation in MRI and the 3D extension of U-Net (Çiçek et al., 2016). For a more general discussion of this literature please refer back to Section 2.5.7.

ConvLSTMs were originally proposed by Shi *et al.* (Shi et al., 2015) for weather forecasting. They have recently been applied to a similar domain to ours in 3D microscopy by Chen *et al.* (Chen et al., 2016b) where they combine their proposed 'kU-Net' with a deep ConvLSTM architecture. In this work they utilise recurrent layers in order to improve segmentations that may be attempted using 2D CNNs. The key differences between the results presented in this chapter and the work of Chen *et al.* (Chen et al., 2016b) are threefold. Firstly, we present results on a novel application, the segmentation of vasculature in multi-photon fluorescence microscopy. Secondly, while they demonstrate the ability of the recurrent networks to improve on the results of a 2D CNN alone, we demonstrate the ability of the recurrent-convolutional hybrid network to learn from targets that are not possible for 2D networks, namely 3D skeletons. In addition to this we explore the necessity of their deep, U-shaped ConvLSTM architecture and compare to a shallow, single layered version.

In addition to the convolutional LSTM, others have suggested the use of orthogonal LSTM units to traverse 3D image volumes, such as in the PyraMiD-LSTM work by Stollenga *et al.* (Stollenga et al., 2015). Other convolutional-recurrent hybrids have also been explored in medical imaging, such as the work by Poudel *et al.* (Poudel et al., 2017), who apply recurrence to the inner layers of a U-Net inspired architecture, for cardiac MRI segmentation.

In this work we will demonstrate the ability of hybrid convolutional-recurrent architectures to approach direct skeletonisation, rather than relying heavily on thinning or tracking algorithms. We believe that by learning to extract centrelines directly in the machine learning task, we remove the degree of subjectivity which is involved in the skeletonisation and pruning pipelines that are popular in many vessel extraction methods.

5.3 Materials

In this chapter we will present experiments based on both the real, preclinical data and our synthetic vascular data. The details for the acquisition and generation

of this data is outlined in more detail in Section 2.1.

For the real experiments our dataset consists of 23 image volumes, of size $512 \times 512 \times 40$. Our evaluation is performed using a 4-fold cross-validation approach with 10% of training data for each fold held back for validation purposes such as early-stopping. Each image is subdivided into tiles of size $128 \times 128 \times 10$, giving each fold approximately 750 training cases and 250 test cases. We keep only 5% of tiles which contain no visible endothelium, so as to avoid excessive class imbalance.

For the synthetic experiments we train on 20 image volumes, of size $512 \times 512 \times 40$, generated as described in Section 2.1. Each volume is subdivided into tiles of size $128 \times 128 \times 16$. Testing was performed on 20 separate volumes, generated identically to the training volumes. All images are normalised to the intensity range $[0, 1]$ before training and testing.

5.4 Methods

Here we will give a description of the network architectures used, as well as details on training and the loss functions considered.

5.4.1 Architectures

CNN: For the CNN portion of the network we use a U-Net style architecture (Ronneberger et al., 2015), identical to that used in Chapter 4, with 2 convolutional layers at each pooling level. Each convolutional layer is followed by a Batch Normalization (Ioffe and Szegedy, 2015) layer and then a (Leaky) Rectified Linear Unit (LReLU) which is defined as follows.

$$LReLU(x; \alpha) = \begin{cases} \alpha x & x < 0 \\ x & \text{otherwise} \end{cases} \tag{5.1}$$

The Leaky ReLU modification attempts to avoid the 'dying ReLU' problem, by maintaining a small slope ($\alpha = 0.1$) in the negative portion whilst retaining the piecewise constant gradient, which provides the efficiency for ReLU activations. We use this as our highly background dominated target data can lead to many 'dead' ReLU units, which we are unable to recover from during training.

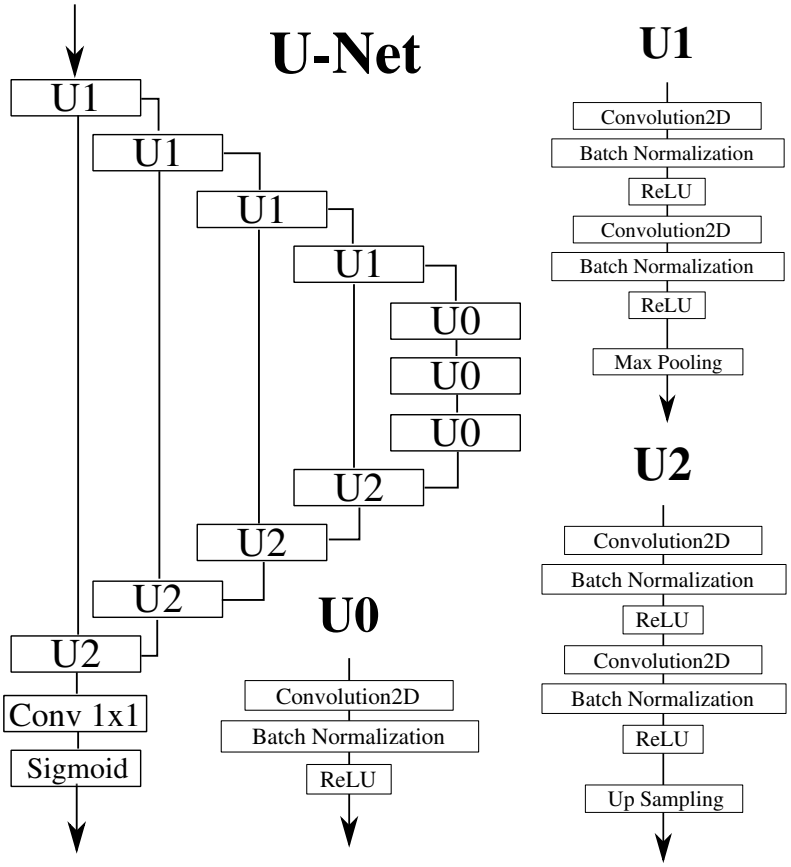


Figure 5.1: Layer arrangement for a each depth layer of the U-Net architecture.

The full architecture is described in Figure 5.1. A sequence of down sampling units (**U1**) provide a broader and broader view on the image. In this coarse-grained view, a sequence of **U0** units extract rich features. These are then up-sampled by a sequence of **U2** units, whilst concatenating these new features with the features from the corresponding scaled **U1** unit.

The final feature map is collapsed using a 1×1 convolution before being passed through a sigmoidal activation. This produces a feature map of the same spatial dimensions as the input image.

All convolutions are performed using 3×3 kernels. Max pooling is performed over a 2×2 region and up-sampling is performed with stride 2. We initialise all weights using a uniform distribution with bounds as suggested by Glorot *et al.* (Glorot and Bengio, 2010).

ConvLSTM: The Convolutional LSTM modifies the traditional LSTM architec-

ture (Hochreiter and Schmidhuber, 1997) by replacing the inner product operations with kernel convolutions, therefore allowing us to efficiently operate on images, as in a CNN. Here we provide the full state equations for this unit and additional discussion can be found in the work by Shi *et al.* (Shi et al., 2015).

$$f_t = \sigma(W_f * x_t + U_f * h_{t-1} + b_f) \tag{5.2}$$

$$i_t = \sigma(W_i * x_t + U_i * h_{t-1} + b_i) \tag{5.3}$$

$$s_t = \tanh(W_s * x_t + U_s * h_{t-1} + b_s) \tag{5.4}$$

$$o_t = \sigma(W_o * x_t + U_o * h_{t-1} + b_o) \tag{5.5}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ s_t \tag{5.6}$$

$$h_t = o_t \circ \tanh(c_t) \tag{5.7}$$

$$y_t = h_t \tag{5.8}$$

Here, $\circ$ represents the Hadamard, or entry-wise, product and $*$ represents kernel convolution. $W_{\{f,i,o\}}$ and $U_{\{f,i,o\}}$ represent trainable weight tensors, x_t represents slice t from the input image stack and y_t represents slice t in the output.

A network schematic version of these equations is shown in Figure 5.2. The input variable x_t and the previous output h_{t-1} are modulated by the weight tensors W_f and U_f respectively, as well as the bias b_f. This value is passed through a sigmoid to produce the value f_t. This is known as the 'forget' gate and will be used to potentially 'zero' or 'forget' some elements of the persistent memory variable c.

The gates i_t and s_t are constructed in a similar fashion. The gate s_t modulates the input and previous output signals via a tanh non-linearity. The o_t gate then dictates which elements of s_t are allowed access to the memory state c_t, via modulation by a sigmoid non-linearity. The output gate o_t dictates which elements of the memory unit can be read off to form the output. The memory unit is modified and passed through a tanh non-linearity. The correct elements are then selected by multiplication with the o_t gate in order to form the output h_t.

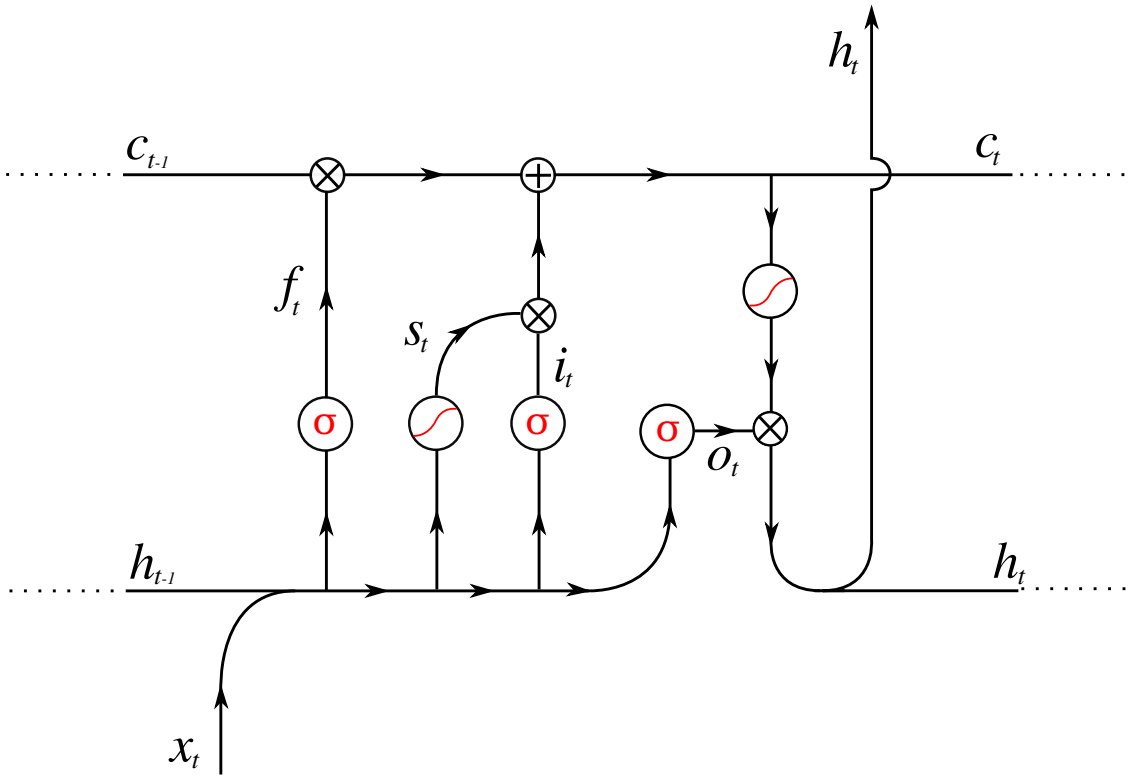


Figure 5.2: Network schematic for the equations given in Eq. 5.8.

We apply each ConvLSTM bidirectionally, concatenating the outputs. Each ConvLSTM layer uses 32 units and kernels of size 3×3. For these experiments we explore this unit applied in both a 'deep' and 'shallow' configuration.

'Deep' configuration: For this configuration we arrange the ConvLSTM layers in a U-shaped configuration, as has been employed by Chen *et al.* (Chen et al., 2016b) in conjunction with a modified kU-Net. We stack two layers of ConvLSTM, followed by a max pooling layer, followed by a further two layers of ConvLSTM, followed by an up-sampling layer, followed by a final two layers of ConvLSTM. The output from this is then passed through a 1×1 convolution to compress the features before a sigmoidal activation to provide the final output. Each ConvLSTM layer consists of 32 ConvLSTM units.

'Shallow' configuration: For this configuration we use a single ConvLSTM layer, consisting of 32 units. A comparison of these two architectures is shown in Figure

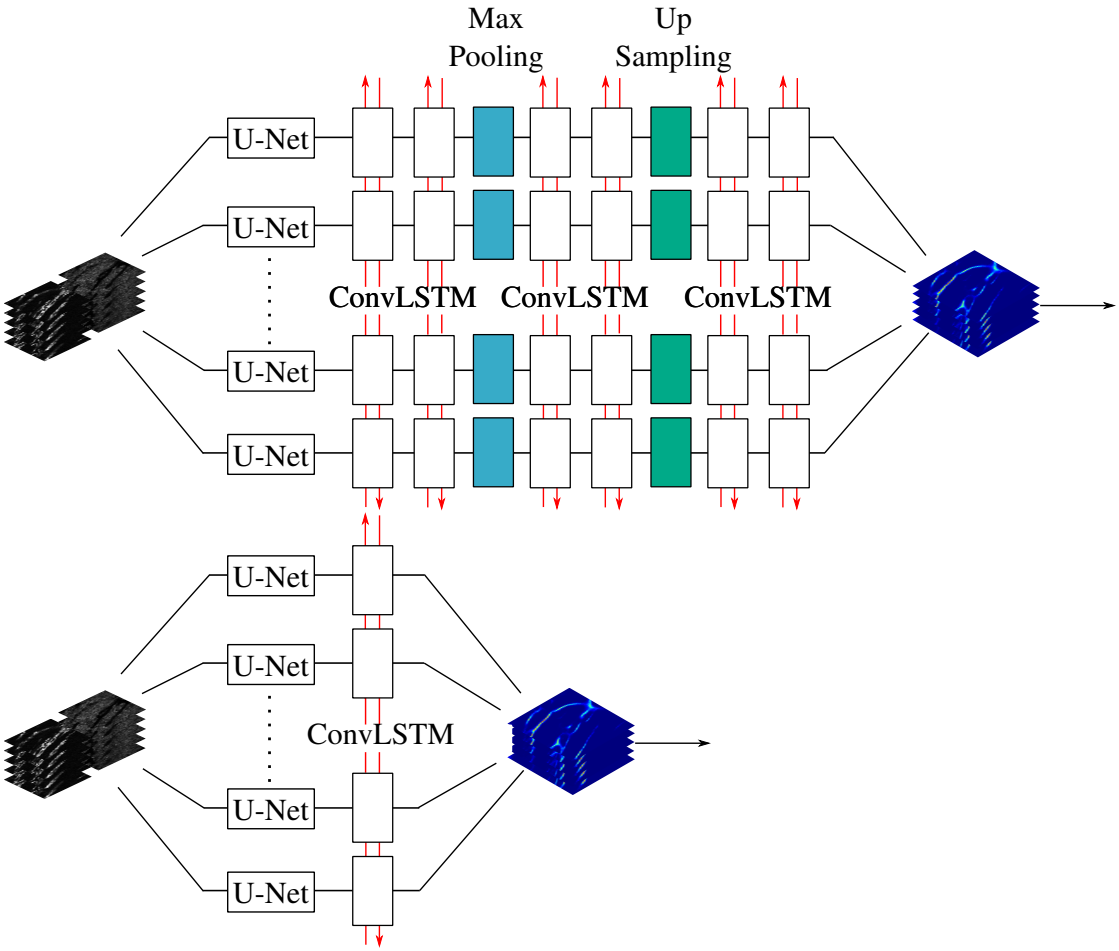


Figure 5.3: Full combined architecture schematic for the U-Net2D + CLSTM networks. Image slices are passed independently through a shared CNN network, before being processed in parallel by stacked bidirectional ConvLSTM units. The final output will be approximately a 3D skeleton which can be refined to give our final skeletal representation.

5.3. Each ConvLSTM configuration is concatenated with a shared U-Net which transforms each image slice independently before passing it to the ConvLSTM.

Combination: In order to combine these networks, we apply a shared copy of the CNN network to each slice of the 3D image volume, in parallel. This produces a 2D skeleton detection on each slice. This new volume is then processed, in series, by the stacked ConvLSTM layers to produce the new, context aware, 3D skeleton detection.

Training: By design, it is possible that this network may be trained in an end-to-end fashion. However, depending on the size of the component architectures, it may

be necessary to separate the training for the two components of the network. In practise we find an 8GB GPU to be sufficient to train even the 'deep' configuration end-to-end, for the image sizes listed in Section 5.3. In order to train the architecture in a split fashion, we suggest first training the CNNs against the vesselness targets of the previous chapter. The outputs from the trained CNN can then be used as inputs for training the ConvLSTM layers, with the 3D thinned segmentations being the targets. All networks are trained using an early-stopping procedure on the validation loss, the epoch with the best validation loss is taken for evaluation.

It is not immediately clear what an effective loss function would be for learning directly from skeletal representations. Here we will compare training against a number of different loss functions. Considering a network output y and target image t we will test a binary cross-entropy loss, given by:

$$\mathcal{L}_{bce}(y,t) = -\sum_i t_i \log(y_i) + (1-t_i)\log(1-y_i). \tag{5.9}$$

A Dice coefficient based loss:

$$\mathcal{L}_d(y,t) = 1 - 2\frac{y \cdot t + \delta}{|y| + |t| + \delta}. \tag{5.10}$$

An advantage of using a Dice coefficient based loss is that it implicitly focuses on the foreground class. This may be advantageous for applications such as this where there exists a severe class imbalance between foreground (vessel centreline) and background. This means that the magnitude of the loss is dynamically balanced to the size of the foreground class in each instance. Here δ is a small positive value to keep the loss well behaved in the case that $|y| = |t| = 0$. For these experiments we set $\delta = 1$.

Another way to address this imbalance, without discarding data, is to use a class-weighted loss function. Where foreground and background observations have their importance weighted according to the ratio of class imbalance.

$$\mathcal{L}(y,t) = -\sum_i (1-\beta)t_i \log(y_i) + \beta(1-t_i)\log(1-y_i), \tag{5.11}$$

where,

$$\beta = \frac{\sum t_i}{\sum t_i + \sum(1 - t_i)}, \tag{5.12}$$

and we use i to index individual pixels within the image. This gives approximately equal importance to both the foreground and background components of the loss function, even though their counts vary significantly. This implies that individual background pixels are assigned a small weight, under the assumption that background is largely homogeneous. However, this has the effect of relatively under-weighting the background regions in the extremely non-homogeneous regions in the immediate vicinity of the thin foreground regions. We therefore propose a pixel-wise loss weighting, similar to Ronneberger *et al.* (Ronneberger et al., 2015):

$$\mathcal{L}_{wbce}(y, t) = -\sum_i W_i t_i \log(y_i) + W_i(1 - t_i) \log(1 - y_i). \tag{5.13}$$

$$W = (1 - \beta) t * g_\sigma + \beta, \tag{5.14}$$

where g_σ represents a Gaussian kernel with standard deviation σ and $*$ represents the kernel convolution operator.

In doing this we incorporate the same degree of class balancing as in Equation 5.11, while spreading its influence to the local vicinity of the thin foreground skeleton. In these experiments we trained our networks using the Adam optimiser (Kingma and Ba, 2015) with a learning rate of 10^{-4}. Although we did not find much sensitivity to σ, the extent of $t * \sigma$ should approximately correspond to the size of the vasculature.

Skeletonisation: A final stage in any pipeline for the analysis of vasculature may include a skeletonisation process, to reduce any kind of detected representation to a topological skeletal representation for further analysis. Uncertainty and variability in our training set can result in skeletonisation results that are thicker than a single pixel wide skeleton, we therefore account for this by performing a skeletonisation as a post-processing. In these experiments we consider a simple homotopic thinning algorithm (Pudney, 1998), followed by a pruning phase to remove artefactual branches. A key advantage of our method is that although we still apply a thinning

algorithm to finalise our results, the majority of the skeletonisation has already been performed by the neural network.

5.5 Results

Here we will present the results of experiments both on synthetic data as well as our real, preclinical microscopy imaging data. We will first briefly outline the architectures used for comparison. We will use the experiments on synthetic data to evaluate a number of the design choices, such as the use of bidirectional recurrent units and the choice of loss function, before comparing the methods on real data.

U-2D: In order to reinforce our assertion in Section 5.1 that a 2D network alone cannot learn from the skeleton targets we include a 2D network for comparison. For this we use a 2D U-Net architecture as described in Section 5.4.1 (CNN). In order to apply this to volumetric images, we apply the network to each slice in the z-stack independently and concatenate the results. Trainable weights: 7,800,000.

U-2D+CLSTM (S): A shared 2D U-Net architecture is combined with a single (shallow) ConvLSTM unit with 32 filters as described in Section 5.4.1. Trainable weights: 8,000,000.

U-2D+CLSTM (D): A shared 2D U-Net architecture is combined with a (deep) U-Shaped ConvLSTM architecture as described in Section 5.4.1, with 32 filters per unit. This design is suggested by Chen *et al.* (Chen et al., 2016b). Trainable weights: 9,100,000.

U-3D: A 3D U-Net architecture, similar to that described in Section 5.4.1 (CNN) but with all 2D convolution and pooling operations replaced with their 3D equivalent. Trainable weights: 18,000,000.

CLSTM: In order to show that the effectiveness comes from the hybrid convolution-recurrent nature of this architecture we include just the ConvLSTM layers alone as a control. For this we use a 'U-shaped' (deep) ConvLSTM architecture as described in Section 5.4.1 (ConvLSTM). Trainable weights: 800,000.

We would like to explore the importance of the rich feature extraction of the CNN architectures (U-2D, U-3D) as well as the impact of the recursive architectures (CLSTM), both with and without a CNN component.

For each experiment we tune an optimal threshold on validation data for each architecture which is used to binarise the output from each network before skeletonisation and analysis.

5.5.1 Results on Synthetic Data

In order to provide a baseline for later comparisons we first compare the ability of our proposed network to a number of comparison networks to segment our synthetic data. We measure this using the Dice overlap, a commonly used segmentation evaluation metric, described in Equation 5.10. We compare a 2D U-Net, our U-Net + ConvLSTM, a ConvLSTM and a 3D U-Net network. All networks are trained to optimise a Dice coefficient based loss, given in Equation 5.10. For evaluation purposes we use the standard Dice coefficient, with $\beta = 0$. These networks are trained against the synthetic segmentation targets. Results can be seen in Table 5.1.

Architecture	Dice Score
U-2D	0.73
U-2D+CLSTM (D)	**0.97**
CLSTM	0.87
U-3D	0.95

Table 5.1: Results for Synthetic Experiment 1

Next we compare the ability of a number of networks to recreate the correct skeletal structure of the vessels in our synthetic data. In order to evaluate this we train all of our networks to recreate the 3D skeletons of the synthetic data, as can be seen in Figure 2.4. We evaluate this using our symmetric Hausdorff average distance as described in the previous chapter. In addition to this we compute the 'coverage' of the ground truth, the fraction of ground truth skeleton that is within some radius of the nearest section of skeleton on the target to be compared. In this case we use a radius of 20μm, as this is comparable in size to the radius of vasculature present in these images.

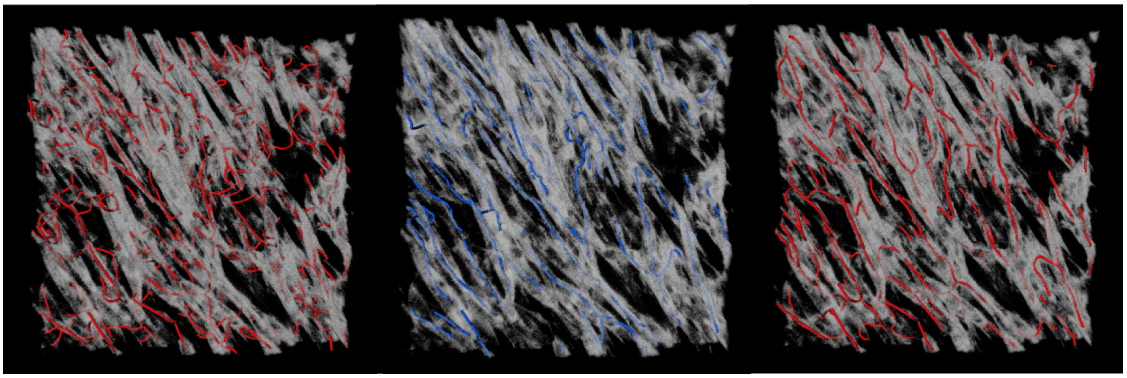


Figure 5.4: Comparison of the U-Net2D method (left) with the proposed U-Net + ConvLSTM approach (right). A manually annotated ground truth is shown in blue (center). Detected vasculature is overlaid onto a volume rendering of the endothelium channel.

Loss function Firstly, we compare the effects of different loss functions for training to reproduce vessel skeletons. The loss functions we compare are the binary cross-entropy, the weighted binary cross-entropy and the Dice coefficient based loss. The architecture used here is the U-2D+CLSTM (D) variant. The results of this analysis can be seen in Table 5.2.

Architecture	Hausdorff (Max)	Coverage
bce (Eq. 5.9)	5.24	0.93
w_bce (Eq. 5.13)	**4.56**	**0.96**
Dice (Eq. 5.10)	n/a	n/a

Table 5.2: Results for Synthetic Experiment 4

In this case we were unable to cause the network to converge on a meaningful solution using the Dice loss function.

Bidirectional Next we evaluate the importance of using a bidirectional ConvLSTM unit, rather than simply running in one direction. A single ConvLSTM unit would only be able to look through the image stack in a single direction, e.g. top-to-bottom or bottom-to-top. This is performed by training a second, independent ConvLSTM layer which is fed a reversed image stack as its input. The output from this unit is then reversed again before being concatenated with the output from the forward layers, in the feature dimension. By applying ConvLSTM units in both

directions and concatenating the outputs we gain the ability to consider changes occurring in both directions. The results of this experiment are shown in Table 5.3.

Architecture	Hausdorff (Max)	Coverage
Bidirectional	**4.56**	**0.96**
Unidirectional	6.47	0.93

Table 5.3: Results for Synthetic Experiment 3

Here we observe an improved performance in the case of the bidirectional network.

Architecture comparison Finally, we compare to our other comparative networks. The ConvLSTM networks are configured in a bidirectional fashion and all networks are trained against a weighted binary cross-entropy loss.

Architecture	Hausdorff (Max)	Coverage
U-2D	78.20	0.07
U-2D+CLSTM (D)	**4.56**	**0.96**
U-2D+CLSTM (S)	7.42	0.88
U-3D	36.33	0.31
CLSTM	41.65	0.31

Table 5.4: Results for Synthetic Experiment 2

From these experiments we observe that the inclusion of networks capable of 3D reasoning is of great importance to this task. In the case of segmentation we see that both the U-2D + CLSTM and the U-3D architecture performed extremely well. However, when we attempted to learn directly from skeletons we found that the U-3D network struggled. The combination of the high Hausdorff score and the low Coverage score show that it was unable to effectively learn the skeleton representation. By contrast the U-2D+CLSTM architectures were able to achieve good results at this task. We also observed that our best results were achieved using Bidirectional LSTMs and our proposed weighting for the binary cross entropy loss function. Finally we see that the CLSTM alone also failed to effectively extract the centrelines. This shows that it is the combination of the convolutional networks with the recurrent networks that is driving the performance.

In the next section we will compare the results using this U-2D+CLSTM architecture to the results from the previous chapter.

5.5.2 Results on Real Microscopy Data

In order to fully quantify the improvements of this approach we will perform a full comparison to the methods of the previous chapter. We will perform this comparison on 21 image volumes and we will train 4 models in a 4-fold cross validation approach. Therefore we will test on all 21 volumes and each model will have been trained on either 15 or 16 image volumes. Image volumes are prepared as described in Section 2.1. The images for this larger training set are taken from 3 different tumours, also outlined in Section 2.1.

We first compare the 'Deep' and 'Shallow' variants of our U-2D+CLSTM combination network. The results of this comparison are shown in Figure 5.5. Although there appears to be a slight improvement for the deep architecture over the shallow one, it is not statistically significant. We therefore prefer the shallow variant as this allows us to use larger volumes during training and has significantly faster execution time (up to 3 times faster).

A full comparison between the methods can be seen in Figure 5.6. In Figure 5.6 we show the distribution of scores under the symmetric Hausdorff average distance as well as p-values for a (two-tailed) Wilcoxon Signed-Rank test between each pair of methods. This is a non-parametric paired significance test. We test significance at a rate of $\alpha = 0.05$, however, as we are performing multiple comparisons we correct using the Holm-Bonferroni correction scheme. Finally we show whether we observe a significant difference between each method according to the adjusted p-value, with blue squares indicating significance. We also present the results of a topological analysis, using the methods developed by Mayerich *et al.* in their NetMets package (Mayerich et al., 2012). Here we compare our suggested T_{max} and G_{max} score. A full explanation and details regarding the computation of these metrics is provided in Appendix A. These results are shown in Figure 5.7.

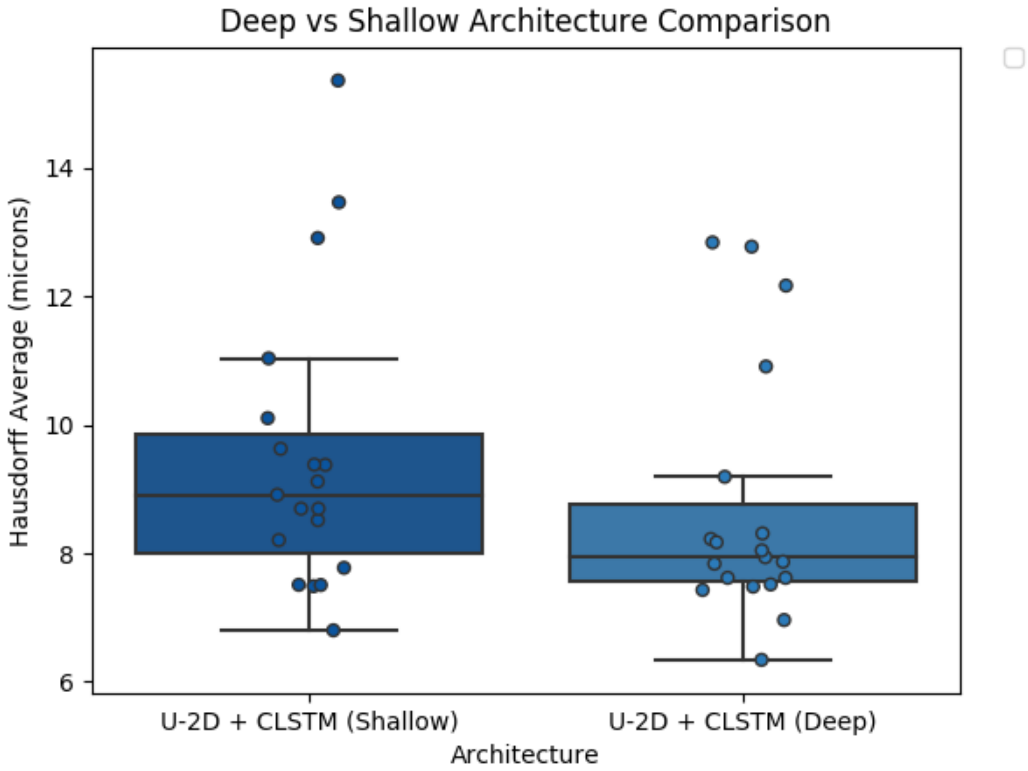


Figure 5.5: Distributions of Symmetric Hausdorff Average for the shallow and deep variants of the U-2D + CLSTM architectures.

In this full analysis we show comprehensively that both the *skel 3d* and *mult 3d* show significant improvement over both previous methods (Wilcoxon Signed-Rank, $\alpha = 0.05$) in the case of the symmetric Hausdorff average, the T_{max} topological score and the G_{max} score. Unlike in the 2D case, we observe no significant difference between the *skel 3d* and *mult 3d* results, showing us that the skeleton network in 3D did not need to rely on the segmentation for robustness.

Interestingly, for this larger dataset we observe less of a significant improvement of the *mult 2d* over *skel 2d*. However, our training set is composed of image sub-volumes from three different tumours. In Figure 5.6 we also show the results of this analysis with the training set split into its constituent groups containing images from each different tumour. We notice that the 2D networks perform significantly better on the simpler cases present in Group 1 and also that *mult 2d* outperforms *skel 2d* in this sub group. In the case of the extremely challenging cases presented

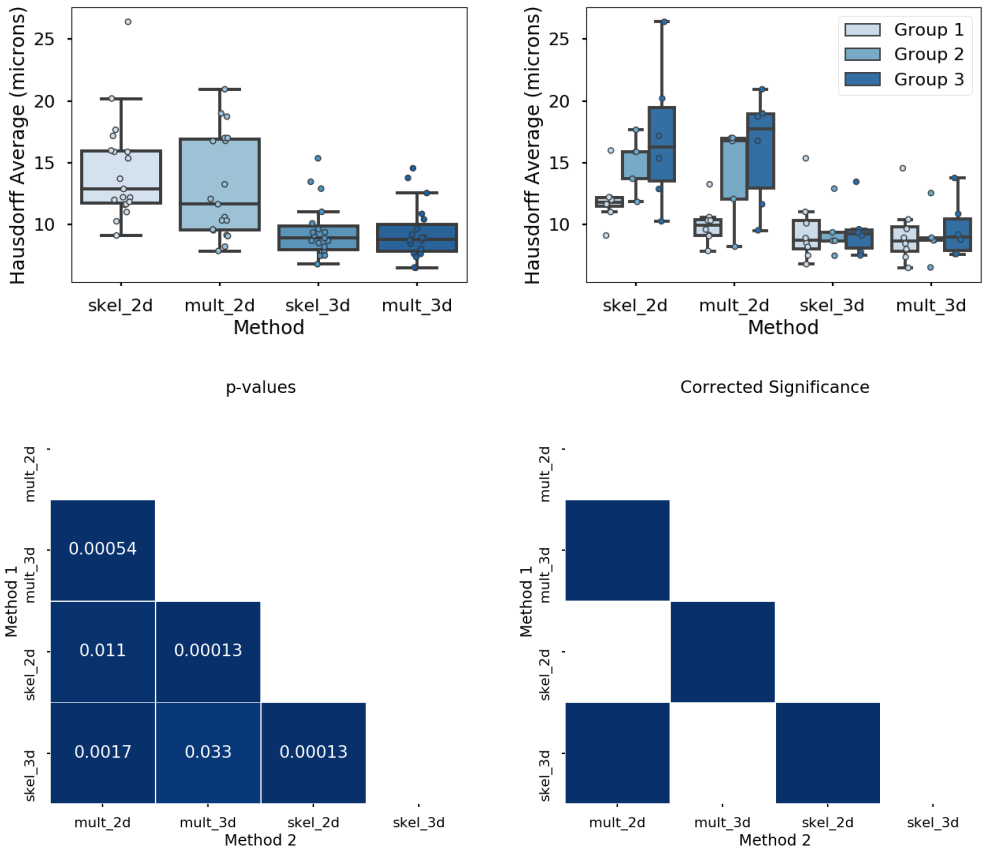


Figure 5.6: Top: Distributions of Symmetric Hausdorff Average for the various competing methods (top left), we also show this distribution for the training set split into its component tumours (top right). Bottom: We show the p-values for a Wilcoxon signed-rank test between each competing method (bottom left). We also show whether this p-value corresponds to a significant difference (Wilcoxon Signed-Rank, $\alpha = 0.05$) after a Holm-Bonferonni correction (bottom right), with blue squares indicating significance.

in Group 3, where differentiation between vessels requires viewing multiple slices of the image, the 3D networks demonstrate their most significant improvement.

5.6 Discussion

In this chapter we have shown that by combining CNN networks with a Convolutional LSTM network, we are able to extract truly 3D vascular structures from complex microscopy images. By doing this, we were also able to avoid the complexity of having to perform convolutions in 3D. We have also shown that using skeletal centrelines as training targets rather than segmentations or our vesselness proxy helps to distinguish between tightly packed structures. In addition to showing

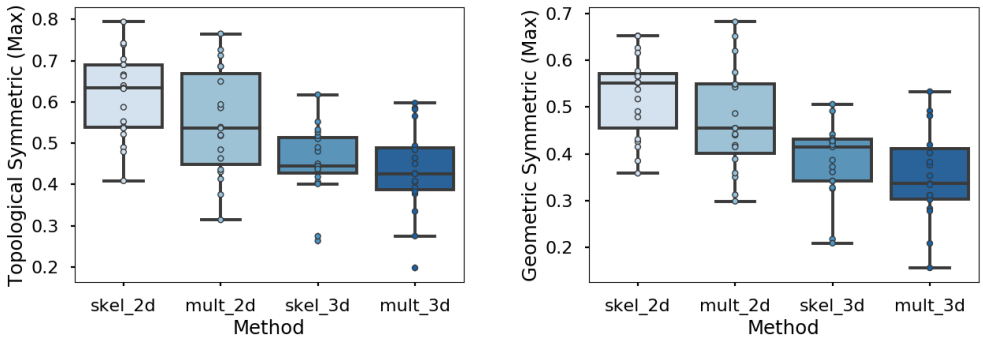


Figure 5.7: Comparison of errors under the T_{max} (left) and G_{max} (right) metrics.

this to be an effective method on our task of extracting vessel skeletons, we have explored the role played by various loss functions, tailored to this task. In our experiments we found that although the 3D CNN achieves comparable results in terms of segmentation overlap, the U-Net + ConvLSTM architecture shows a clear improvement in terms of its ability to recreate skeletal representations of the vasculature, with the 3D CNN failing completely in the case of the real data. A qualitative comparison of the difference in results between the U-Net2D and the U-Net2D + ConvLSTM architectures can be seen in Figure 5.4. In the results from the 2D network we notice that a large number of spurious connections have formed between non-connected parts of the vasculature. We interpret this as being due to the networks inability to separate structures in 3-dimensions. The 2D network performs adequately in regions of low density (for example along the top edge of the image) however begins to fail in the regions of high density in the center. However, the 3D approach performs well in all of these regions, as we are able to localise and separate responses in 3D.

In our comparison of the loss functions, given in Table 5.2, we observe that the use of a weighted binary cross-entropy loss produced the best results. While the Dice derived loss function may perform well for segmentation and performs a kind of implicit class balancing, we found that it was not capable of learning a meaningful model for these extremely sparse skeleton structures. We hypothesise that this may be due to the fact that the Dice coefficient tends towards binary solutions, as discussed in Section 4.1.2. The cross-entropy loss was able to learn

that the 'likelihood' of the centreline was maximised in a ridge along the vessel. However, due to the complex and subjective nature of the exact location of the centreline, this likelihood was low everywhere. Since the likelihood was low, the Dice coefficient may tend towards a zero solution, making it unsuitable for vessel extraction. Further weighting of the Dice coefficient, beyond the implicit foreground class balancing, has been explored recently by Sudre *et al.* (Sudre et al., 2017) and Fidon *et al.* (Fidon et al., 2017) with similarly imbalanced classes ($\sim 1/10000$), which may go some way towards rectifying this. However, we believe that the sparsity of our structures, rather than the clusters present in their images (white matter lesions), may exacerbate this problem. Additional discussion on the role of the Dice coefficient as a loss function can be found in Section 4.1.2.

Further to this analysis of the algorithmic components, we have demonstrated the significance of this 3D approach over the 2D methods of the previous chapter using both a geometric measure, the modified Hausdorff distance, as well as a set of topological measurements. In both cases we observed significant improvements for the 3D methods, demonstrating that the inclusion of 3D information is critical, not only for accurately representing the geometry of the vasculature, but also for correctly modelling the connections present in the network. In terms of the topological measurements, shown in Figure 5.7, we observed the best results from our *mult*3*d* network combination. We attribute this to the skeleton networks improved ability to localise vasculature (thus reducing the false positive rate) combined with the segmentation networks ability to reinforce the signal at junctions (reducing the false negative rate). We found the most common failure mode in the 2D networks to be an extremely high false positive rate. This could be attributed to the networks inability to correctly localise vasculature due to relying on a proxy for the 3D structure.

We believe the results achieved by our *skel 3d* and *mult 3d* networks are extremely good representations of the vascular networks. We have assessed the inter-rater reliability of manual skeletonisation for this data and consider it to be approximately 6μm under the symmetric Hausdorff (max) measure and approximately 0.25 in the case of both the T_{max} and G_{max} measurements. This highlights the complexity of

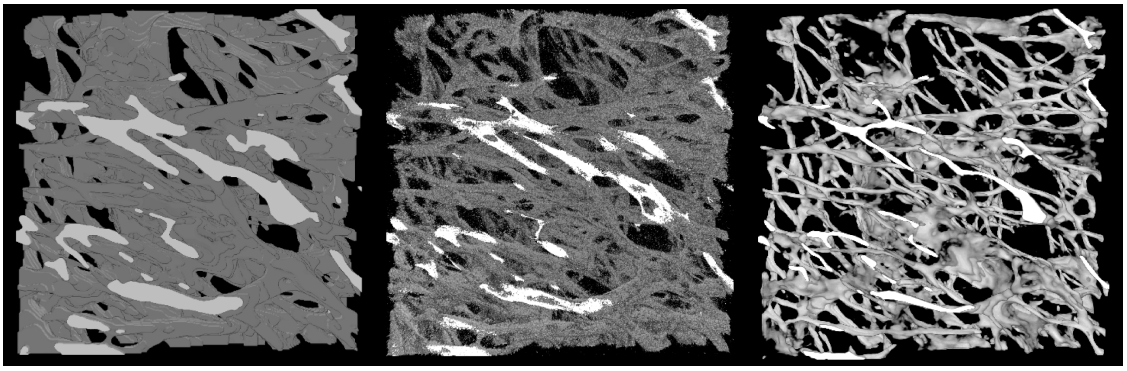


Figure 5.8: Network trained to reproduce segmentation (left), input endothelium channel (center) and network trained to reproduce skeleton (right).

the task at hand, but also should emphasise that the methods developed here have gone a long way toward achieving human level delineation for these images.

In addition to these results, it is useful to examine to relative model sizes of the architectures being compared. In the case of the synthetic architecture comparison experiment, we see that the U-Net2D was completely incapable of learning this 3D representation. This is to be expected as contextual 3D information is required. However, the addition of a single ConvLSTM layer, resulting in an increase of about 2.5% in the number of trainable weights, makes this representation possible. By contrast, the natural extension of the U-Net2D to the U-Net3D requires at least an approximate tripling in the number of parameters, as 3×3 kernels are extended to $3 \times 3 \times 3$ volumetric kernels. Although a smaller 3D network could be considered and thus require fewer weights, we know that the 2D U-Net of these dimensions works well on our 2D data and we would ideally like to retain this power. We found that while the U-Net3D architecture performs equivalently in the segmentation task, and is arguably simpler to train, it was unable to meaningfully learn from our skeleton target data.

Convolutional-recurrent architectures of this kind represent a sort of hybrid between 2D and 3D image processing. This is fitting for data of this kind as it also represents a hybrid between 2D and 3D imaging, where images acquired predominantly in 2D are concatenated to form image volumes. Interestingly, we noted that in the case of the real data, the deeper, U-shaped ConvLSTM architecture

proposed by Chen *et al.* (Chen et al., 2016b) did not significantly out-perform our shallow single ConvLSTM unit architecture. Although this may change in the presence of larger training sets, we believe that this is an important observation, as networks that are able to perform with smaller training sets are of great interest to researchers in biomedical applications, where large public datasets will not usually exist for bespoke applications. We believe that the most pertinent observation is that the addition of a single ConvLSTM layer, a 2.5% increase in the number of trainable weights, was transformative in terms of the original CNN's ability to model 3D data. This is contrasted to the original work of Chen *et al.* (Chen et al., 2016b), in which they demonstrate a marginal gain on a task for which a regular 2D U-Net produces respectable results.

In addition to this, the smaller architecture exhibits considerably lower memory consumption during both training and execution, allowing larger image tiles or batch sizes to be used. We also observe a considerably faster run time with the shallow network evaluating 2-3 times faster than the deep network.

We believe that the pipeline we have demonstrated here represents a principled approach to information sharing, that targets the majority of the computational power of the network to the primary source of information (convolutions in-plane) and forces efficiency in the use of trainable parameters. We envision this approach also being useful in other anisotropic imaging modalities where 3D structures are extracted from a concatenation of higher resolution in-plane images, e.g. cardiac MRI, lung CT or other microscopy applications. We also believe that training the network to reproduce only the centrelines rather than the entire segmentation is an extremely effective method for vessel segmentation in tightly packed vascular networks. A qualitative demonstration of this is shown in Figure 5.8.

We also acknowledge a number of limitations with this method which we will discuss here. Firstly we observe issues with convergence of the CNN+ConvLSTM unit, that is resistant to standard initialisation techniques such as Xavier initialisation (Glorot and Bengio, 2010) or orthogonal kernel initialisation. This is also resistant to the use of Leaky ReLU to avoid the 'dying ReLU' problem, wherein

all activations of a ReLU unit lie in the zero region of the function for the entire training set and thus will never escape. In approximately one quarter of training runs we observed the network falling into a pathological local minima of the 'all-zero' solution, which is caused by the extremely sparse nature of the dataset. It is this that our weighted loss function attempts to prevent, but it still happens on occasion. A simple but effective solution to this seems to be to initialise the weights of the CNN section of the network with weights trained on either the segmentation or the vesselness network described in the previous section. Once the network has escaped the pathological attractor of the all-zero solution, convergence proceeds without issue. A second limitation is that we still suffer from the diffuse nature of our output in regions of uncertainty. In regions where the network is not certain about the presence of vasculature, we will often end up with a diffuse, weak signal. This causes a heavily reliance on the choice of threshold for binarisation of the results, in order to recover the correct topology. This is due to the fact that the 'skeleton' representation we are attempting to learn possess a conditional structure. This means that the value of one pixel in the output may affect the validity of the value another pixel takes. This is not something that we can learn directly via pixel-wise loss functions alone. It is this limitation that we will attempt to address in the next chapter.

6

Refining 3D Skeletons with Conditional Generative Adversarial Networks

Contents

6.1 Introduction 142
6.2 Methods 144
6.3 Experiments 148
6.3.1 Circle-Square Example 148
6.3.2 2D Vessel Example 152
6.3.3 Real Data 155
6.4 Discussion 159

6.1 Introduction

Although in the previous chapter we have successfully learnt from the skeleton targets directly, we still suffer from a degree of locational uncertainty. This can originate from the variability in manual annotations, but may also derive from the general subjectivity attached to something like a 'centreline' for such irregular objects. When training in a supervised manner, using cross-entropy or some variant, we are implicitly asking the network to learn the function *"What is the likelihood that this pixel lies on a vessel centreline, given the input image?"*. This may be insufficient for two reasons: firstly, the location of the vessel centreline may be ill-posed and contains a degree of subjectivity. It is particularly subjective for large vessels. This means that our predictions will be weaker for larger vessels, not because they are more difficult to detect but simply because the exact location of the centreline is more subjective. The second deficiency is that the concept of a 'centreline' possesses conditional structure. This means that although we may be able to regress something approximating the 'likelihood of being on the centreline' we cannot simply sample from these pixel-wise distributions independently and receive a valid centreline. The pixels that compose this skeletal representation are conditional on the value that their neighbours take. This leads us to suspect that directly learning skeletal representations from neural networks is not something that can be done using supervised pixel-wise loss functions alone.

Generative Adversarial Networks (GANs) were proposed and popularised recently by Goodfellow *et al.* (Goodfellow et al., 2014). In this work they showed that by formulating the training of a network as an adversarial minimax game between a generator and a discriminator network, it is possible to train the generator network to learn to sample from a target space without ever being presented directly with samples from the space. Further discussion of adversarial training for generative networks is provided in Section 2.5.6. In work by Radford *et al.* (Radford et al., 2015) they demonstrated the possibility of using de-convolutional networks as the generator network (DCGAN) that was able to learn to sample from the space of natural images. The advantage of this over comparable methods such as Variational

Auto-Encoders (VAEs) trained on a pixel-wise reconstruction loss is that GANs do not suffer from the smoothing of the space caused by optimizing an average loss over the training sets (Kingma and Welling, 2013). This results in sharper and more realistic generated natural images. GAN training was further extended to allow for supervised training, in the form of conditional GANs (Mirza and Osindero, 2014; Zheng et al., 2015; Denton et al., 2016; Isola et al., 2016). In this formulation the GAN is learning to sample from the space of mappings from input to target image, given a set of supervised training pairs. This means that our network is able to learn to reproduce higher order relationships between input and output that transcend the pixel independence that other state-of-the-art loss functions, such as pixel-wise categorical cross-entropy, suffer from.

GANs have notably been quite opaque in their training process, as improvements to the generator are only observed relative to the discriminator and vice versa, rather than comparing against some static validation set as in the case of traditional supervised learning. This can lead to instability during training and make it difficult to know in what sense the generator network is approaching some kind of 'optimal' solution. Although theory on networks of these kinds is currently sparse, progress is being made in this area, for example with the Wasserstein GAN (WGAN) formulation of GAN training (Arjovsky et al., 2017). This work by Arjovsky *et al.* claims improved stability through the use of an Earth Movers distance as opposed to the binary cross-entropy used in the original formulation for the adversarial loss component. The cross-entropy may be problematic as it has a zero gradient outside of the support of the target distribution. They also claim that the losses from this distance function are more 'interpretable'. However, in this chapter we will train using the standard GAN protocol (Goodfellow et al., 2014), due its established position in the literature.

Although GANs have largely been used in the generation of synthetic data, some interesting applications in medical imaging have emerged. Kamnitsas *et al.* (Kamnitsas et al., 2016a) have applied adversarial learning to the task of domain-adaptation, improving robustness to different imaging protocols. By setting up

an adversarial training scheme whereby the discriminator attempts to determine whether the image comes from the initial training domain or the desired target domain (by considering the internal activations of the generator) they force the generator the learn domain invariant features. Through this approach they are able to adapt their model to new imaging domains without the need for additional annotated training data. Wolterink *et al.* (Wolterink et al., 2017) have employed GANs in order to address noise issues associated with low-dose CT. They train a conditional GAN in order to generate synthetic regular-dose CT images from low-dose images in an attempt to perform a de-noising. Recently, Lahiri *et al.* (Lahiri et al., 2017) have applied adversarial training to vessel segmentation. In this work they generate synthetic training data for retinal fundus segmentation and show that by doing this they are able to achieve comparable results to the state-of-the-art using considerably fewer real training images.

In this chapter we will demonstrate how the inclusion of a conditional adversarial loss allows our network learn to effectively sample from the space of valid centrelines directly rather than regressing some vesselness quantity. This can also be interpreted as a form of domain-adaptation, as we wish to transition from the space of vesselness likelihood functions, to the space of valid vessel centrelines. We will first describe the framework used to achieve this and then demonstrate the effectiveness on a number of synthetic examples as well as our real data. To our knowledge this is the first example of adversarial training being utilised for direct skeletonisation.

6.2 Methods

Architecture: The 'generator' architecture we will utilise here, in the case of 3D data, consists of a shared 2D U-Net architecture (Ronneberger et al., 2015), with 4 down-sampling layers, and two layers of convolutions at each depth layer. This shared U-Net is stacked with a Convolutional LSTM layer (Shi et al., 2015; Chen et al., 2016b). In the previous chapter we have shown the effectiveness of such an architecture in extracting 3D skeletal structures in anisotropic image volumes. In the case of the 2D examples we will use a standard U-Net architecture as the 'generator'.

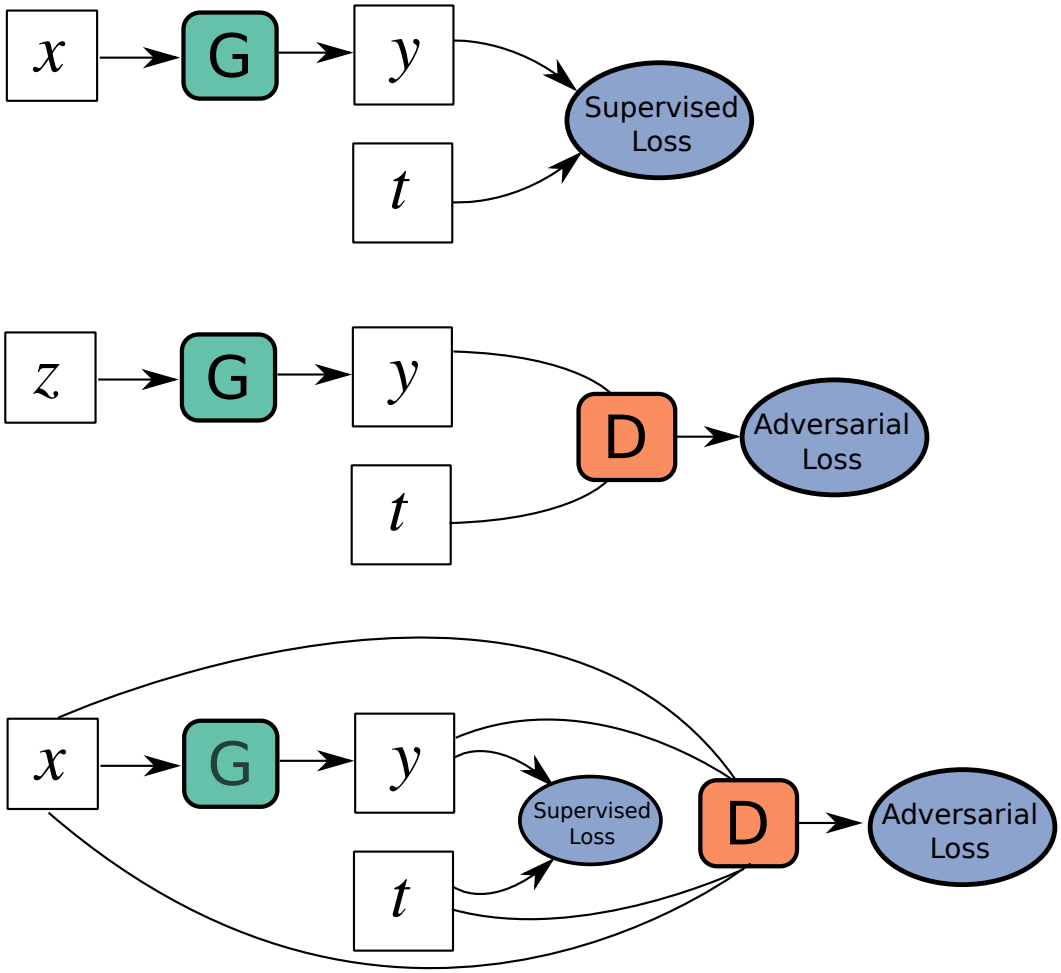


Figure 6.1: Comparison of network schematics for supervised, generative adversarial and conditional generative adversarial networks. Top: Traditional supervised learning of the network G given training inputs x and training targets t. Middle: GAN training of the unsupervised training data t, seeded by random noise z. Bottom: Semi-supervised conditional GAN training of the network G given training inputs x and training targets t.

The 'discriminator' architecture that we will use follows a standard design, taking the advice provided by the DCGAN work of Radford *et al.* (Radford et al., 2015). It consists of repeated application of strided convolutions with 5×5 kernels until we have reached the desired level of down-sampling. Convolutions are applied with a stride of two giving a down-sampling of $\frac{1}{2}$ for each convolution. We will utilise the same architecture for the discriminator network in both 2D and 3D however, for the 3D data we will use 3D kernels. Due to the high anisotropy of this data we will only down-sample in the x and y directions, even in the case of 3D data.

Adversarial Training: In this chapter we will employ the framework of Mirza *et al.* (Mirza and Osindero, 2014) in training a conditional GAN. Similar to the work of Isola *et al.* (Isola et al., 2016), we will apply this framework to a fully convolutional generator network, conditioning on our input images for both the discriminator and the generator. Let $(x, t) \in X$ be a pair of input/target images. Our discriminator now assesses the validity, conditional on the input image, $D(t|x)$. This is achieved by concatenating the image volumes of t and x in the feature dimension to produce the input of the network D.

$$\min_G \max_D V(D, G) = \mathbb{E}_{(x,t)\sim p_d} \left[\log(D(t|x)) + \log(1 - D(G(x))|x)\right] \tag{6.1}$$

It is this formulation that we will use to train our adversarial skeletonisation network. In a traditional supervised training scheme, we attempt to minimise some pixel-wise loss. This means that even though we may apply some spatial prior within the network, our output pixels are conditionally independent in terms of the loss. In this case the discriminator attempts to learn the mapping from input image to output image. The generator is then forced to learn this mapping, which could contain higher order dependencies between the output values of pixels. As was suggested by Isola *et al.* (Isola et al., 2016), in order to guide the training in a broad sense, we will add a supervised loss, ℓ_{sup}, for the generator as well, giving:

$$\min_G \max_D V(D, G) = \mathbb{E}_{(x,t)\sim p_d} \left[\log(D(t|x)) + \log(1 - D(G(x))|x) + \ell_{sup}(G(x), t)\right]. \tag{6.2}$$

The L_1 norm is commonly proposed for image reconstruction, however we will use the binary cross-entropy, in keeping with earlier chapters. With this in mind we are able to break down equation 6.2, including a supervised term, into the following two minimisation problems, $\mathcal{L}_d$ and $\mathcal{L}_g$:

$$\mathcal{L}_d = \frac{1}{|X_b|} \sum_{(x,t)\in X_b} \lambda \left[\log(D(t|x)) + \log(1 - D(G(z))|z)\right] \tag{6.3}$$

$$\mathcal{L}_g = \frac{1}{|X_b|} \sum_{(x,t)\in X_b} \lambda \ell_{adv} + (1-\lambda)\ell_{sup} \tag{6.4}$$

$$\ell_{adv} = -\log(1 - D(G(x))|x) \tag{6.5}$$

$$\ell_{sup} = -\sum_i t_i \log(G(x_i)) \tag{6.6}$$

We also introduce the parameter λ to govern the influence of the supervised training versus the adversarial training, without loss of generality. Here X_b denotes some mini-batch of the training set X, and λ is a tuning parameter adjusting the relative weights of supervised and adversarial training. We can see that the semi-supervised conditional GAN framework is an elegant combination of supervised training and adversarial training, allowing us to model both pixel-wise and more complex relationships between our input images and our target images. A schematic illustration of the differences and similarities of these training paradigms is shown in Figure 6.1. After training, the network G may then be used on its own as a predictive network, identically to one trained with standard supervised learning.

In work by Isola *et al.* on image-to-image translation (Isola et al., 2016), they propose a range of different styles of discriminators, for use with fully convolutional generators. In this work they outlined three styles of discriminator network, the ImageGAN, the PatchGAN and the PixelGAN. Given an input image of size $N \times M$ the ImageGAN returns a variable from the space

$$d \in (0, 1), \tag{6.7}$$

which gives an over-arching decision on fake/real for the entire image. The PatchGAN returns

$$d \in (0, 1)^{\bar{N}\times\bar{M}}, \tag{6.8}$$

where $\bar{N} = 2^{-k}N$ for some number of down-sampling steps k. This represents a separate fake/real decision for 2^{2k} disjoint patches of the image. The PixelGAN returns

$$d \in (0, 1)^{N\times M}, \tag{6.9}$$

and attempts to provide a fake/real decision for each pixel in the input image.

In order to derive an ImageGAN style discriminator from a PatchGAN style discriminator for these experiments, we apply a global max-pooling operation to the patches in the PatchGAN before passing the result through a single sigmoid activation. We have also considered a fully connected layer to learn a linear model for the patches as well as a global average pooling layer, but we found the global max-pooling layer to be most effective.

6.3 Experiments

6.3.1 Circle-Square Example

In order to demonstrate the problem of structured outputs and motivate a solution, we set up a toy example wherein our network is presented with an image of either a circle or a square and is tasked with placing a smaller shape within this larger shape. A cross in the case of the circle and a smaller square in the case of the square. The larger shapes are placed randomly within the confines of the image and the smaller shapes are placed randomly, within the extent of their larger shape, in the target image. Examples of this data are shown in Figure 6.2. This can be considered to be a simulation of annotating subjective or qualitative targets in object detection. Although this may be an exaggerated imitation of the subjectivity associated with image annotation, we believe it highlights an important point regarding the ability of standard training protocols.

If we consider training this task in a purely supervised manner, what we will learn is the likelihood, for each pixel, of containing part of the target shape, conditioned on the location of the input shape. This means that we will learn to approximate some combination of the distribution of the jitter and the shape of the cross. In this case, that should be approximately a uniform distribution within the circle. In the case of very small amounts of jitter, we would expect to be able to recover an accurate reproduction of the cross targets. This is because what we are aiming to reproduce is a 'conditional distribution', meaning that pixel values in the output are not independent of one another. The location of

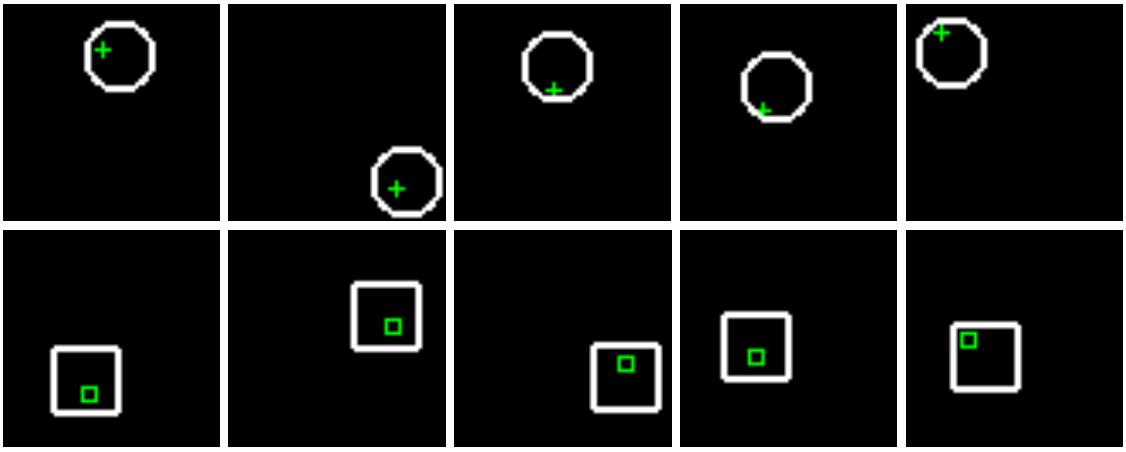


Figure 6.2: Examples of the input images overlaid with their respective target images. Input data is the large circle or square (white) and the training target is the small cross or square (green).

the central pixel in the cross, entirely constrains the locations of the other pixels of the cross. However, this is not something that we can encode with a simple pixel-wise loss function like cross entropy.

In this experiment we observe the differences between learning with and without an adversarial or 'generative' loss. For the supervised case we train against a binary cross-entropy loss

$$\mathcal{L}_{bce} = -\sum_i t_i \log(y_i), \tag{6.10}$$

$$= -\sum_i t_i \log(G(x_i)). \tag{6.11}$$

For the adversarial training we train a conditional GAN as described in Section 6.2 with a combined semi-supervised adversarial loss as described by Equation 6.6. We optimise this as suggested in the original work by Goodfellow (Goodfellow et al., 2014), alternating training $\mathcal{L}_d$ and $\mathcal{L}_g$ for an equal number of examples each. As in previous experiments we use the Adam optimiser (Kingma and Ba, 2015), with a learning rate of 0.0001. In this case we set $\lambda = 0.1$, meaning the supervised training should largely dominate the adversarial until the supervised training converges. We find that even with the supervised term dominating the adversarial term as in this case the network, training is still substantially influenced by the adversarial aspect of the loss. The strong influence of the supervised learning plays a role

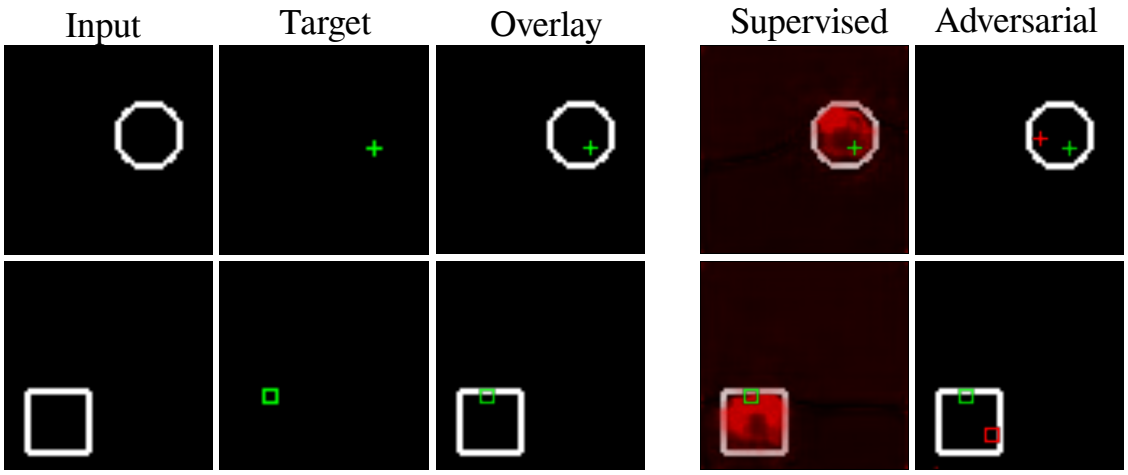


Figure 6.3: Qualitative comparison of the outputs from from the supervised and the semi-supervised adversarial networks. Input images are shown in white, training targets are shown in green and network predictions, given the respective input images are overlaid in red.

in stabilisation of the training, especially in the early stages of training when a weak discriminator network may not be able to provide a sufficiently informative gradient. Due to the difficulties currently associated with quantifying the training of GANs, we rely on a heuristic approach to select λ. We find that values in the range $[0.01, 0.1]$ appear to provide the most stable results, whilst still allowing solutions distinct from the purely supervised minima.

It is clear to see from Equations 6.6 that the supervised training scheme is equivalent to setting $\lambda = 0$. For this task we found the ImageGAN style of discriminator to be most effective.

A qualitative comparison of the results of this experiment is shown in Figure 6.3. From this we see that see that only the adversarial network was able to correctly learn from our jittered targets. As expected, the supervised network reproduces a diffuse probability map of where the pixels might be, whereas the adversarial network has learnt the qualitative sense of what our training set represented. The adversarial network is able to produce 'candidate' images that represent samples from the conditional distribution that the supervised network has learned. Of course, it should be noted that which of these two outputs is 'better' is entirely subjective. Information on the uncertainty and spread of these objects (as learned

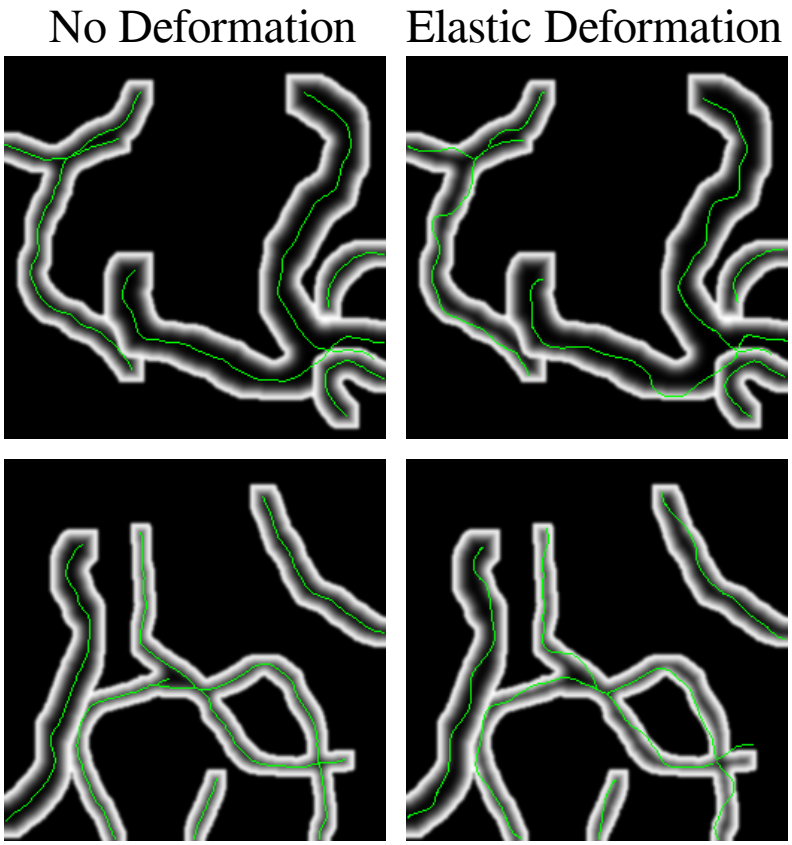


Figure 6.4: Two Synthetic vessel structures (white) with overlaid skeletons (green). Left: regular centreline. Right: with elastically deformed centreline.

by the supervised network) may also be desirable, depending on the task at hand.

This example is interesting as it represents a microcosm of the problems encountered in the previous chapters where we attempted to learn from a structured output (the skeleton) and were limited by our ability to only learn the diffuse likelihood function. We were then forced to reconstruct the skeleton by further processing such as thresholding and skeletonisation which are not necessarily informed by our training data and, depending on the method of skeletonisation, may be either prone to adding additional errors or may be time consuming to compute. Although many sophisticated skeletonisation and tracking algorithms exist, we see value in attempting to incorporate this entire process into a single end-to-end network.

6.3.2 2D Vessel Example

As discussed previously, a key real world instance of this kind of 'qualitative' target is in the subjectivity of annotating certain structures in images. When training a network to detect vasculature it is significantly more efficient for the annotator to simply sketch the centreline of the vessel rather than segmenting the entire thing. However, the centreline of the vessel may be fairly subjective, especially in 3D, which can make manual annotation very time consuming and difficult. In addition to this subjectivity there will also be a degree of irreducible inaccuracy induced by the line-drawing ability of the annotator. Here we test the ability of our network to learn vessel centrelines from annotations that have been distorted or jittered in order to replicate this problem. We generate our synthetic vessel-like images using the protocol described in Section 2.1.2, however, for these images we restrict ourselves to 2D and do not add noise.

Here, our 'elastic' distortions are created by generating two images, consisting of uniform random noise $u, v \sim \mathrm{U}(-\alpha, \alpha)^{n \times m}$. These images are then both smoothed with Gaussian kernels, g_σ, to create the x and y components for a smooth deformation field. Our skeletons are then re-sampled according to this deformation field. An example of two vessel structures, shown with both a regular and a deformed skeleton overlaid, is shown in Figure 6.4. For these experiments we use values $\alpha = 100$ and $\sigma = 10$. We train using 100 images of size 512×512 and test on a set of 100 unseen images of equal size.

Although skeletonisation is a discrete process and, as we have discussed, skeletons are difficult to learn via pixel-wise loss functions, we present some ideas regarding the encoding of skeletonisation directly into supervised loss functions in Appendix C.

As observed with the Circle-Square dataset in Section 6.3.1, the supervised method is forced to learn the uncertainty jointly with the appearance model, which leads to a more dispersed signal. However, in this case, due to the more complex task, this leads to uncertainty as to the skeletal structure. As the network can not learn that the skeletons should be continuous, this may lead to topological 'breaks' in the structure. Also we see that at regions of uncertainty such as branching points the

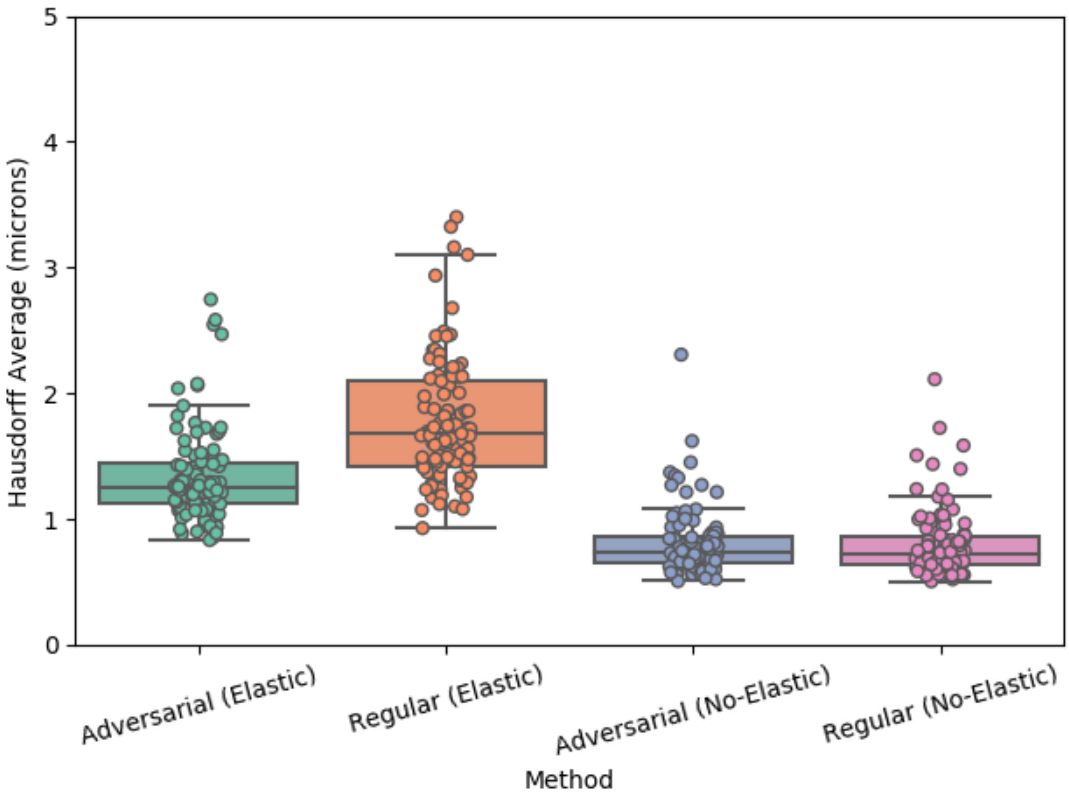


Figure 6.5: Comparison of Hausdorff average between Adversarial training on elastic deformations (left), regular training on elastic deformations (centre) and regular training without deformations (right).

signal becomes significantly more diffuse. Although an ability to report uncertainty is valuable, here we would like to separate the uncertainty from the proposed vessel centreline. In order to explore this we present a quantitative comparison of these methods, by considering the Hausdorff average between these predicted skeletons and the true, non-deformed ground truth skeleton.

We wish to assess the role played by adversarial training in the case of this deformed training data. In order to investigate this we train a U-Net2D network, similar to that used in Chapter 4, both adversarially ($\lambda = 0.9$) and regularly ($\lambda = 0$) and both with and without elastic deformations to the training targets. This gives us four networks: Adversarial (Elastic), Adversarial (No-Elastic), Regular (Elastic) and Regular (No-Elastic). For this task we found that a PatchGAN style discriminator ($k = 3$) was the most effective.

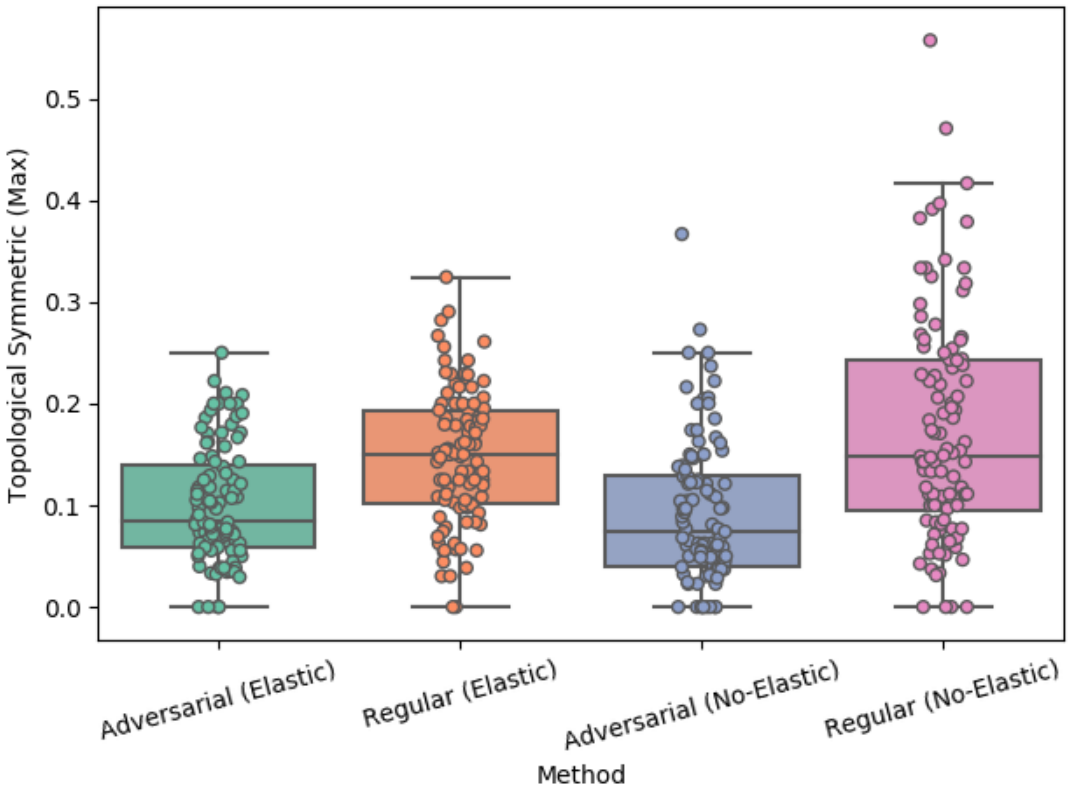


Figure 6.6: Comparison of the topological T_{max} score between Adversarial training on elastic deformations (left), regular training on elastic deformations (centre) and regular training without deformations (right).

Results

We present the results for each method, optimally thresholded and skeletonised, in Figure 6.5 and Figure 6.6. As we can see from the Hausdorff Average results in Figure 6.5, the task of reproducing these centrelines in the case of non-deformed training data is achievable to a very high standard by both the Regular (No-Elastic) and Adversarial (No-Elastic) methods. However, the inclusion of elastic deformations in the training data severely degrades the performance, as seen in the Regular (Elastic) method. By including an adversarial loss when training against the elastically deformed training data, as seen in the Adversarial (Elastic) method, we are able to go some way toward recovering the performance exhibited in the case of the non-deformed training data.

Additionally, we performed a comparison of the topological agreement using

the T_{max} measurement of previous chapters. The results of this comparison are shown in Figure 6.6. Interestingly, we see that although the Regular (No-Elastic) method scored significantly (Wilcoxon Signed-Rank, $\alpha = 0.05$) better than the Regular (Elastic) method in the case of the geometric Hausdorff measurement, we found no statistically significant difference between them in the case of the topological T_{max} measurement. Additionally, we see that both the Adversarial (Elastic) and Adversarial (No-Elastic) methods significantly (Wilcoxon Signed-Rank, $\alpha = 0.05$) out-performed both of the Regular methods under this measurement. These results also hold for the G_{max} measurement.

A qualitative comparison of the network outputs for the Adversarial (Elastic) and the Regular (Elastic) networks is shown in Figure 6.7. Here we see, as discussed in Section 6.1, that the adversarial network was indeed able to learn the structured, thin skeleton representation of the training data, even despite the elastic deformations. The regular training learns a more diffuse signal, similar to that seen in Section 6.3.1 (although less extreme), due to the lower levels of uncertainty in the training data.

6.3.3 Real Data

Finally we demonstrate that this approach can be used to refine the vessel extraction approach developed in Chapter 5. For this section the generator network takes the form of the hybrid convolutional-recurrent CNN-ConvLSTM network, using the 'shallow' configuration. The discriminator network takes inspiration from the DC-GAN network of Radford *et al.* (Radford et al., 2015) replacing 2D convolutions with 3D volumetric convolutions but only down-sampling or pooling in the x-y plane. Although we avoided 3D CNN networks for handling our input image, due to the nature of the images as z-stacks, we believe that we are justified in using a 3D CNN for this purpose. This is because our output images more closely represent a true volume image as we have trained against skeletons which are truly contiguous in 3D and do not suffer from slice misalignment. Discriminator networks are also, typically, significantly smaller in terms of model capacity than their respective generator network. This may go some way towards ameliorating our other concern

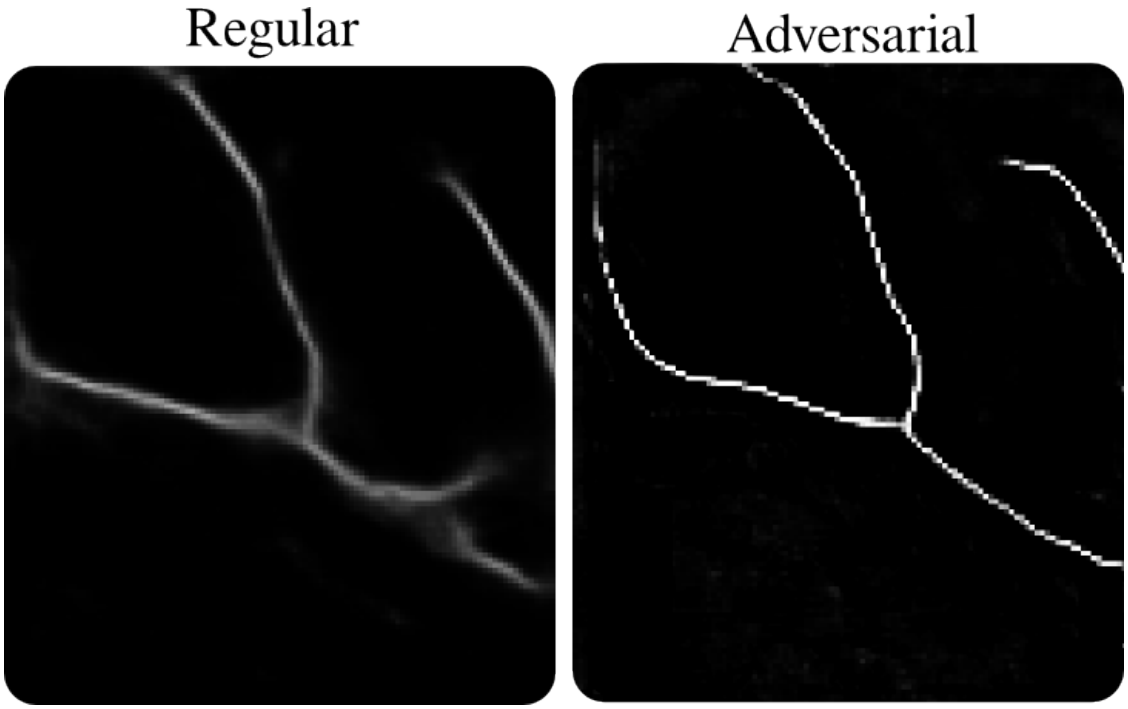


Figure 6.7: Qualitative comparison of the network outputs for Adversarial (Elastic) and Regular (Elastic). We see that the thin skeleton structure has been learnt by the adversarial approach whereas the regular approach has learnt a more diffuse likelihood, incurring a break.

regarding 3D CNNs. The optimisation we perform is identical to that shown in Equation 6.6, with the exception that we replace ℓ_{sup} with our Weighted BCE loss from the previous chapter, given in Equation 5.13. After training, the generator network G will be compared against the equivalent architecture, trained with a regular training protocol, from the previous chapter. We train and test using the same data as in Chapter 5, splitting 23 image volumes into a 4-fold cross validation.

A comparison of the network output both with and without adversarial refinement is shown in Figure 6.8. We can clearly see that the properties of the 'skeleton' are best retained by the adversarial output, which proposes valid centrelines rather than a diffuse vessel likelihood. Line breaks in this image can largely be attributed to the proposed centreline continuing on an adjacent image slice. The difference is most striking in the case of large vessels. Although large vessels may be the easiest to detect, as discussed earlier there is larger subjectivity associated to the location of the centreline. Two particularly problematic examples of this are

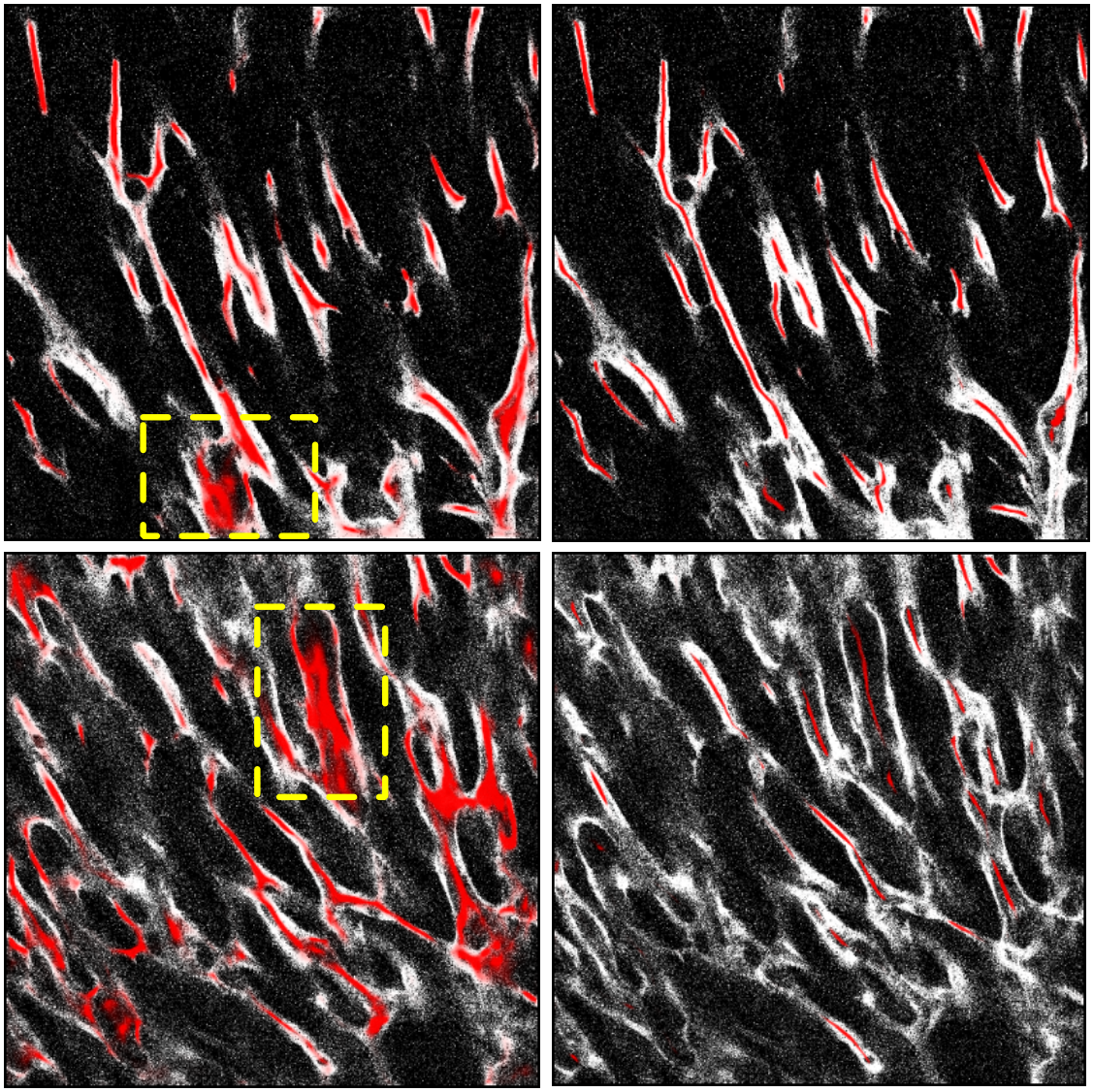


Figure 6.8: Comparison of Adversarial training (right) against purely supervised training (left) for learning vessel centrelines. The top and bottom rows show two different slices through the same image volume.

highlighted in Figure 6.8. In both of these cases the vesselness likelihood is locally multi-modal, even though there is only a single vessel. This may lead a method of skeletonisation, via thresholding or otherwise, to conclude that there are multiple vessels present. The adversarial method, however, has presented a single centreline. For this task, as in the 2D vessel simulation we found the PatchGAN ($k = 3$) to be the most effective discriminator architecture.

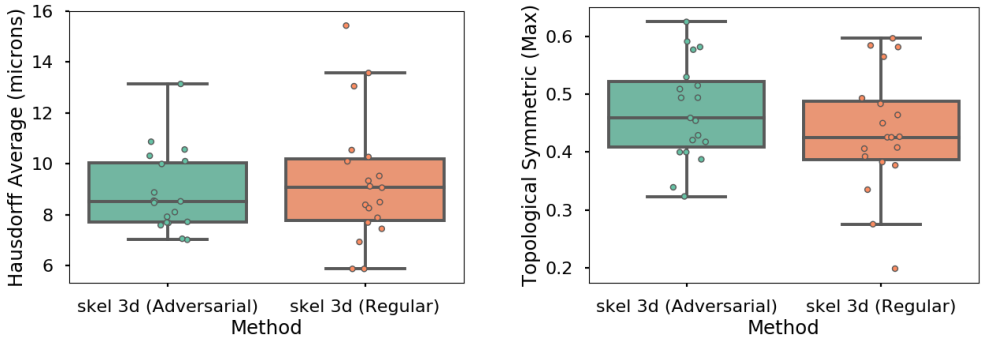


Figure 6.9: Left: Comparison of the optimal symmetric Hausdorff average for the regular and adversarial training methods. Right: Comparison of the optimal T_{max} score, described in Section 5.5.2.

Results

As discussed, our main aim with this process is to reduce our dependence on optimally thresholding in order to achieve good delineations. In order to assess this we will test for our symmetric Hausdorff average distance score when thresholding our output at all possible points over the range of intensities. A method that is not dependent on optimally choosing the threshold should achieve a good score over a large range of the possible thresholds, whereas a highly dependent method may only achieve a good score at very specific thresholds. We will compare this on our ConvLSTM method of Chapter 5 and the adversarially refined improvement of it proposed here. We will also compare the optimal score achieved in both this Hausdorff analysis as well as the topological method used in the previous chapter (Mayerich et al., 2012) for both methods, the results of which are shown in Figure 6.9.

As can be seen in Figure 6.9 we observe no statistically significant change (Wilcoxon Signed-Rank, $\alpha = 0.05$) between the optimally thresholded scores for both methods. However, in Figure 6.10 we observe the key contribution. We see that for the adversarial method, this optimal score is achieved almost regardless of the choice of threshold, whereas in the case of the ConvLSTM skeleton method of Chapter 5 we are heavily dependent on the optimal threshold. As this optimal threshold is not known *a priori* for a new dataset, we believe this to be a very valuable contribution.

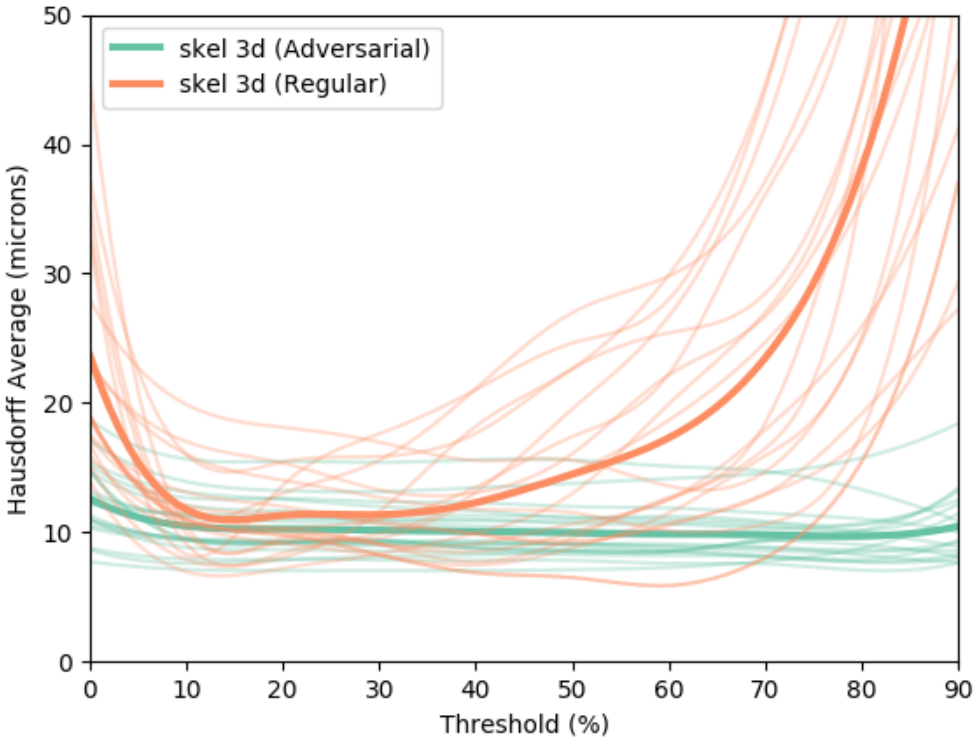


Figure 6.10: Traces for the symmetric Hausdorff average thresholded over the entire range shown for each test image. Each image in the training set is represented by a trace for each method. An average trace for each method is shown in bold.

Although here we have used this approach to train a generator model G from scratch, we achieve similar results in a shorter time by initialising the adversarial training using a generator that has already converged after regular training, as described in Chapter 5. Although this kind of training may result in problems for regular GAN training, due to the mismatch in ability between generator and discriminator, we find that the heavy use of the supervised loss function ℓ_{sup} makes this largely irrelevant.

6.4 Discussion

In this chapter we have shown that conditional adversarial training provides a powerful method for learning to reproduce sparse structured targets like skeletons.

In our first toy example, we showed that if our target contains a highly structured

output, in addition to being conditional on the input, then we will struggle to learn both of these processes using regular pixel-wise supervised learning alone. We observed that while the regularly trained network was able to learn to output a response within the confines of the input shape, it was not able to learn that an input circle corresponds to an output cross, and that an input square corresponds to an output square. Contrasted to this, we see that the inclusion of an adversarial loss term allows the network to learn all of the relationships within the training set. This may appear to be a trivial or arbitrary task but we believe that it represents the issues observed later, in microcosm.

We then moved closer to our true application with our second toy example, where we attempted to learn the centrelines of some simplified, 2D *vessel-like* structures. This is an almost trivial task in the regular case, however, we observe the effects of distorting the target centrelines by some elastic deformations. Similarly to the Circle-Square example, we see that this induced uncertainty causes significant harm to the regular supervised networks ability to delineate the centrelines correctly, as the uncertainty is increased and we observe a more diffuse signal. Interestingly, we see that this deterioration in the geometric Hausdorff measurement did not translate into a significant deterioration in the topological T_{max} measurement. This makes a degree of intuitive sense as our elastic deformations were a geometric transformation, which preserved the topology. We also observed that both adversarial method produced a superior topological prediction of the test data when compared to both purely supervised methods. This suggests that adversarial training may be beneficial when attempting to learn representations that wish to preserve topology.

Finally, we have tested this approach as a refinement to our CNN+ConvLSTM architecture from the previous chapter. We found that we were able to qualitatively improve the appearance of centreline delineation, either by refining a previously trained architecture or training one from scratch using this procedure. The advantage of this is most clearly seen in the case of large vessels which suffer the most from the imprecise delineation given by supervised learning (as shown in Figure 6.8). In our quantitative experiments we found that we were able to eliminate our

dependency on choosing an optimal threshold without degrading performance whatsoever from the previous case. In Figure 6.10 we see that the adversarial method achieves a score comparable to the optimal score in the regular case for almost the entire range of thresholds. We confirmed this by showing that the optimal scores for both methods did not show a statistically significant differences (Wilcoxon Signed-Rank, $\alpha = 0.05$) in either the Hausdorff method or the topological method, T_{max} (Mayerich et al., 2012).

An interesting observation over all of these experiments is that the ImageGAN discriminator was optimal for the Circle-Square task, but the PatchGAN discriminator was optimal for the 2D and 3D vessel tasks. This seems to make intuitive sense as the Circle-Square task had a single focal point in each image whereas the focal points in the vessel images are diffuse and numerous. This means that it makes more sense to reduce the Circle-Square task images to a single fake/real decision, whereas in the vessel examples we may wish to take a differing view over different parts of the image. It is our contention that the diffuse response observed in the Circle-Square toy example from the purely supervised method, seen in Figure 6.3, is analogous to that seen in Figure 6.8 on the real data. We believe that the same mechanism that allowed us to learn the structure of the squares and crosses allows us to avoid the diffuse, uncertain responses in the real vascular images, and ultimately perform more robust delineations, as seen in Figure 6.10. We believe this to be the first use of GANs to learn skeletonisation directly and that this approach has great potential in improving how we learn to delineate vasculature and other elongated structures, beyond segmentation.

Although we believe this to be an extremely promising method, there remains a number of limitations. Firstly, this is an extremely memory intensive training protocol as we are required to back-propagate through the composed network $D(G(x))$. This means we are unable to maximise either the size of our generator network or the batch size, as may be desired with regular training. Secondly, although we observe a generally good topological score, we observe that one of the main failure modes our adversarial networks is that they may struggle to correctly

join junctions in some cases. This is due to the fact that the vast majority of the training data consists of the branches between junctions rather than the junctions themselves. In this case we suggest a gap-filling algorithm may be used as a post processing step to remedy these small junction breaks. Previous work, proposed by ourselves (Bates et al., 2015), suggested a tensor-voting based gap-filling framework targeted at contrast enhanced vasculature. This may be suitable here as there still exists a residual 'signal' at the junction, it is just diminished. This is broadly analogous to the low levels of residual contrast agent found in stenosed tumour vasculature that this method was designed for. Additionally, if desired, neuron tracing algorithms may also be applicable as a post-processing step here.

A major issue surrounding all adversarial networks currently is the stability of training. Although the supervised component adds a great deal of stability, mechanisms such as early stopping cannot be employed effectively. This is because, by design, we are not aiming for the optimal supervised loss. Therefore we are unable to use this as a mechanism to prevent over-fitting, as our adversarially trained network represents some kind of guided over-fitting to begin with. This means that a manual early stopping mechanism must be employed with a human observer viewing some validation sample at each epoch and choosing which epoch looks 'best'. This is clearly far from ideal, however many authors are working on this problem of meaningful adversarial loss functions. Although we are unable to demonstrate an improvement in terms of the optimal scores, we believe that significant improvements could be possible via further refinement of the discriminator architecture and training protocol.

7

Observations, Discussion and Outlook

Contents

7.1 Longitudinal Vessel Analysis 164
7.2 Summary 173
7.3 Discussion 175
7.4 Directions for Future Research 179
7.4.1 Adversarial Learning for Skeleton Extraction 179
7.4.2 Unify into a Multi-Task Learning Framework 180
7.4.3 Further Network Analysis 180

In the previous chapters we have developed a number of methods, representing a sequential improvement in our ability to extract the vascular networks from images of tumour vasculature in multi-photon fluorescence microscopy. However, there still exists a gap between the image processing task and our ability to derive useful information from these images. In order to further demonstrate the value of these approaches, we will outline the full pipeline that we employ, enabling an analysis of real experimental data.

In this chapter, we first outline our pipeline for vascular network feature extraction. We then present a longitudinal analysis that has been enabled by the previously developed methods. Next, we summarise and discuss the findings of this thesis and their significance to this application. Finally, we explore a number of avenues that we believe would merit further investigation in continuing this work.

7.1 Longitudinal Vessel Analysis

In this section we will briefly present an analysis that has been enabled by these methods, on real experimental data. We will investigate changes in the vasculature that occur longitudinally during daily imaging of tumours both with and without angiogenic or vascular disruptive agents administered. We will perform analysis of a number of quantitative measurements taken from the tumour vasculature.

The tumours that we will analyse come from three different classes:

1. **Control:** Untreated tumours.

2. **DC101:** Tumours treated with the DC101 antibody. This should block VEGFR2 signalling, reducing the sprouting of vessels. This in turn should reduce the number of branches and 'normalise' vasculature.

3. **Dll4:** Tumours treated with an anti-Dll4 antibody. This should induce high rates of branching in the tumour vasculature, known as 'hyper-sprouting'.

All tumours are taken from the MC38 cell-line and consist of 3 cases from Group 1 (Control), 4 cases from Group 2 (DC101) and 3 cases from Group 3 (Dll4). Efforts

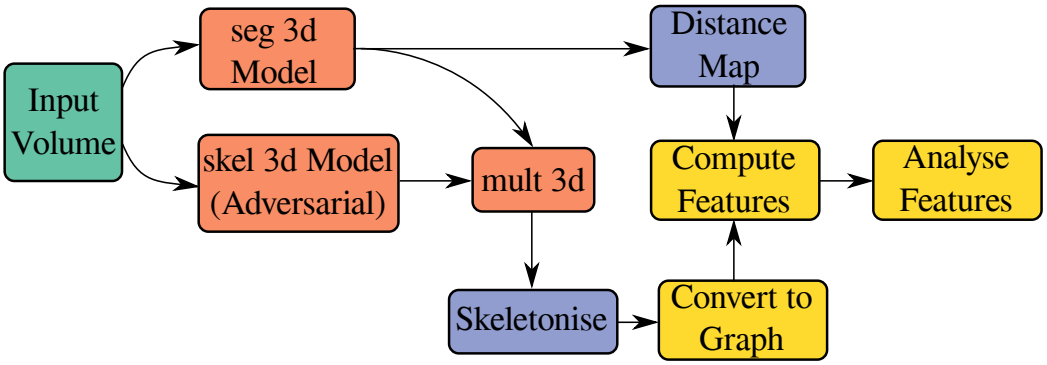


Figure 7.1: Schematic for full segmentation and analysis pipeline. The segmentation model as described in Chapter 5 and the adversarially refined skeleton model as described in Chapters 5 and 6 are combined by taking a geometric average, as explored in Chapters 4 and 5. We estimate vessel diameters from the accurate segmentation model and extract the network topology from the skeletonised *mult 3d* image. The skeleton network is then traversed to form a discrete graphical representation of the network and features are attached to the edges and nodes of this graph. Analysis can then be efficiently performed over the entire vascular network.

were made to begin treatment on days when the tumours were at a similar size, this is denoted day 0 for all cases. Imaging begins at day 0 for all tumours with the exception of one case from the DC101 group for which imaging began one day before treatment, leading to a measurement for day -1.

A full schematic for the extraction and analysis pipeline is shown in Figure 7.1. For all tumours we perform a full segmentation using both the *seg 3d* and *skel 3d* models, which are combined to form the *mult 3d* combination. We then perform skeletonisation and pruning on this image to reach a discrete skeletal representation. Vessel diameters are estimated using *seg 3d*. The network topology is determined by traversing the discrete skeleton image in order to form a graphical representation of the network, where the edges consist of branches and the branch points are the nodes. We attach morphological and image based features to each edge in this graph and locational information to each node. For the purposes of presentation, in this chapter, we project our feature maps into 2D. Although we could calculate all metrics in full 3D, we believe that for ease of presentation we are justified in

projecting into 2D, due to the fact that the extent of the images in the x-y plane is considerably larger than that in the z direction.

Alongside more traditional measurements such as branch length, vessel diameter and tortuosity, we also investigate some network level properties such as directional coherence and branch point density. We suggest the following definition of directional coherence. Consider a branch b as consisting of a number of points $\{p_1, p_2, \ldots, p_n\}$. We can approximately summarise the direction of this branch by the unit vector $u = \frac{p_n - p_1}{||p_n - p_1||}$. Given a point in space and some radius r we collect a set of branches $\{b_1, b_2, \ldots, b_m\}$ that are within the radius r of our chosen point in space. We can summarise this point by computing the average dyadic product tensor,

$$T = \frac{1}{n} \sum_{i=1}^{m} u_i \otimes u_i. \tag{7.1}$$

Now if we consider the eigenvectors of this tensor T

$$\lambda_1, \lambda_2, v_1, v_2 = \text{eig}\,(T)\,, \tag{7.2}$$

we are able to extract an orthonormal basis aligned along the principal directions within T. The eigenvalues give a measure of the importance of each direction. If $\lambda_1 = 1$ and $\lambda_2 = 0$ then the vectors $\{u_1, \ldots, u_n\}$ were entirely aligned in direction v_1. If $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$ then the vectors $\{u_1, \ldots, u_n\}$ contained no single dominant direction. We therefore construct a measure of coherence, similar to that used by Bates *et al.* (Bates et al., 2015):

$$\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}. \tag{7.3}$$

This gives a value of 1 if our local vectors were highly aligned and a value of 0 if they were disordered. Computing this at regular points across our tumours we hope to identify regions of order and disorder. However, we acknowledge that such a definition may not be sufficient to differentiate between boundaries of perpendicular but isotropic regions and highly anisotropic regions. We compute branch point density using kernel density estimation (KDE) on the locations of the branch points. We can use a normalised KDE to quantify how branch points

are distributed within the tumour and a non-normalised KDE to estimate branch point density at every point in the tumour.

An example of the outputs from the various steps of segmentation, skeletonisation and analysis are shown in Figures 7.2 and 7.3. Using these methods of measurement, we make a number of observations and demonstrate that these observations correspond to our understanding of the mechanisms at play.

First we observe the average density of vessels. We compute this by taking the average of the non-normalised kernel density estimation maps for the branch points, over the domain of the tumour. In Figure 7.4, we see the results of this measurement. We have presented both the raw values, reported in branch points per μm^2 as well as a normalised measure, where each measurement is divided by the measurement at Day 0, to account for the natural inter-tumour variability.

In these results we see that we are able to clearly measure the drastically increased density caused by the 'hyper-sprouting' behaviour induced by the anti-Dll4 antibody. By considering the normalised measurement we see that on average the Dll4 treated tumours increased in density considerably more than the control tumours from the point of initial treatment. We also observe that the DC101 tumours exhibited a consistent decrease in density after treatment, aligning with our understanding of the 're-normalisation' effect.

Next we observe the average length and diameter of the vessels. The results of these comparisons are shown in Figure 7.5. In terms of average branch length, we observe that the Dll4 treated tumours exhibit a decrease in average branch length after treatment, whereas the DC101 treated tumours exhibit growth, above that of the control group. In the average diameter measurement, we see that growth of vessel diameter appears to be stunted in the case of the Dll4 tumours, however, by observing the non-normalised data it seems that the Dll4 cases examined happened to have larger initial diameters, which may contribute to this observation. Although all tumours are from the same cell line, general non-homogeneity and variety within each tumour may account for slightly varying initial conditions. We also see that, in the non-normalised data, the DC101 exceeds the control and Dll4 tumours. In

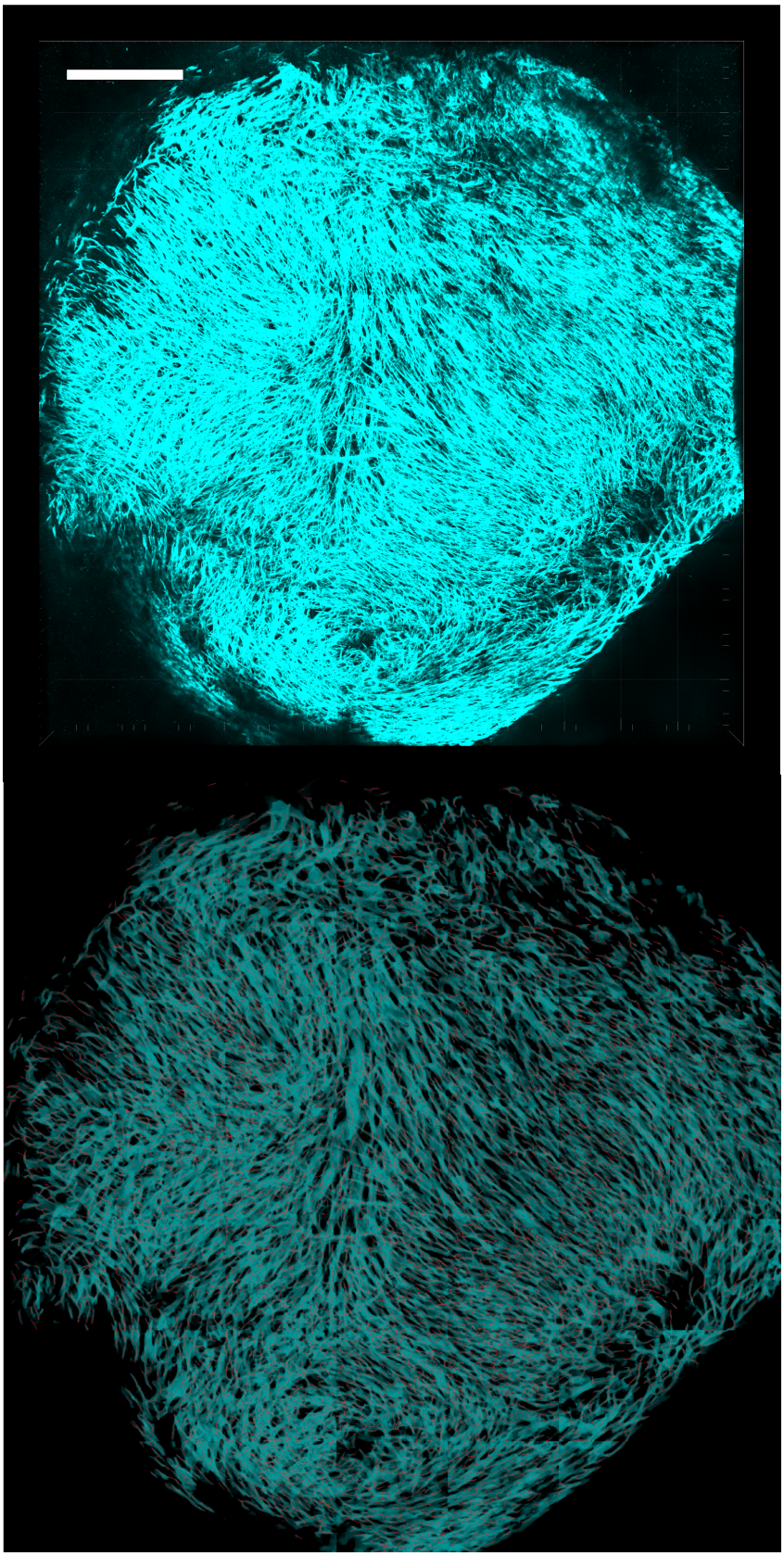


Figure 7.2: Comparison of input image with segmentation and skeletonisation output.

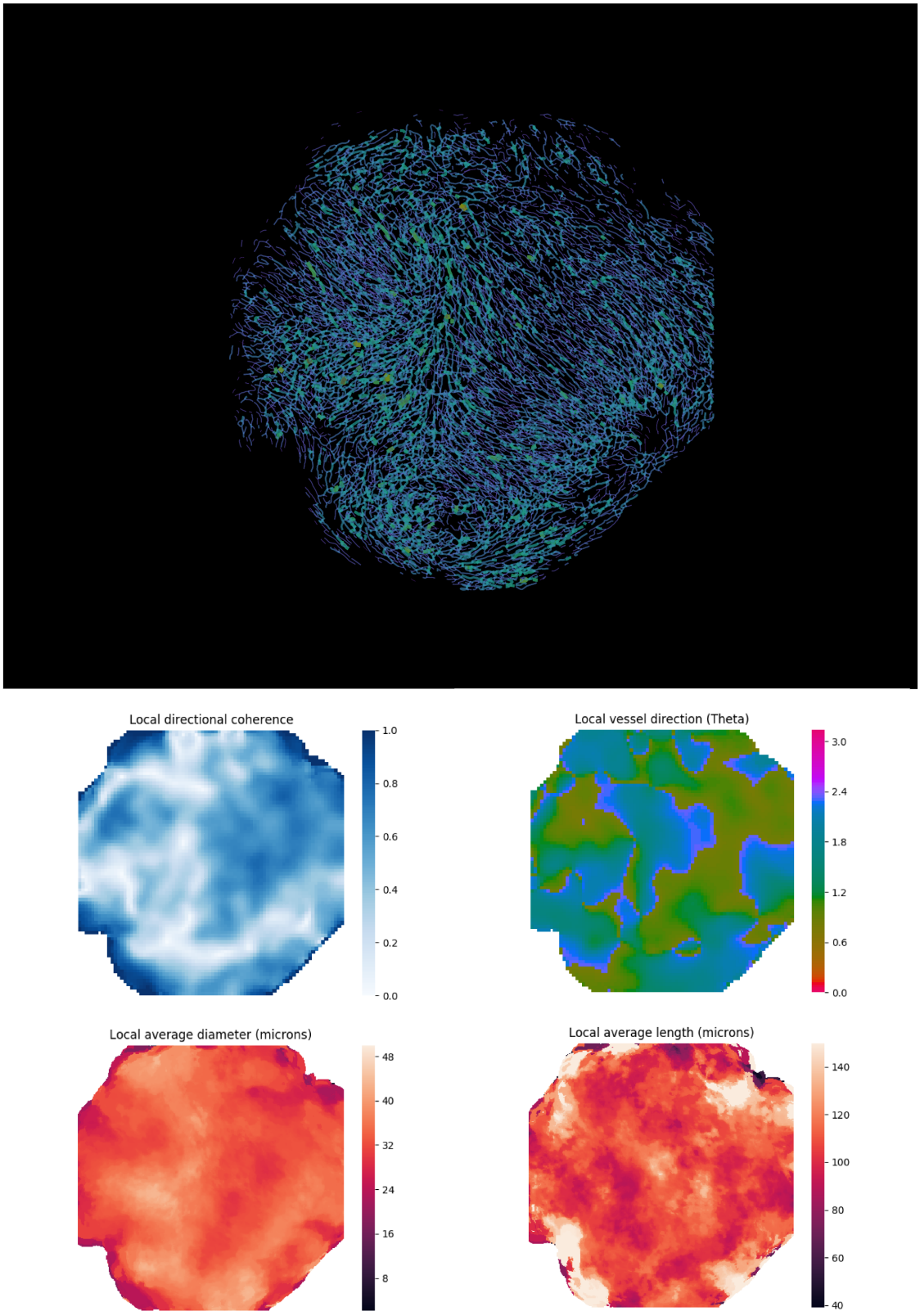


Figure 7.3: Extracted feature graph of the entire tumour (top). Various local averages of features (bottom).

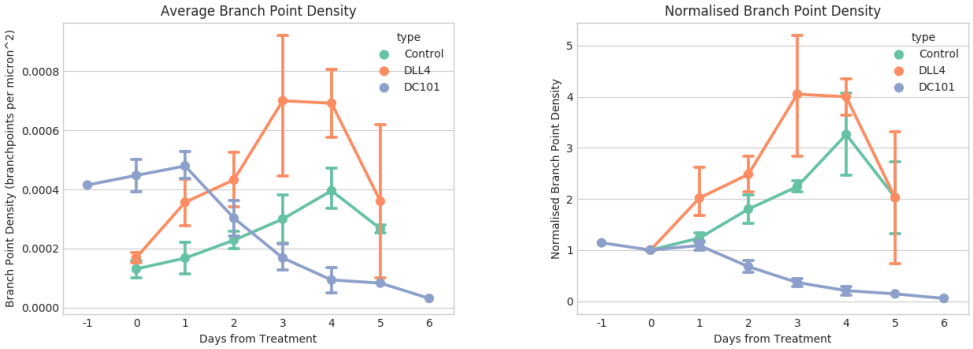


Figure 7.4: Changes in vessel branching density shown against days since treatment. Results are shown for the raw values (left) and the normalised values (right). Error bars show a 68% confidence interval for the mean.

order to further investigate the cause of this we examine the relative prevalence of 'large' vessels in each tumour case. For this purpose we define large vessels as those with a diameter greater than 100µm. In Figure 7.6 we see the results of this analysis. It is clear that the frequency of these large vessels is significantly higher in the case of the DC101 tumours, which is the driving force behind the increasing average diameter for this tumour class. We see that in both count and proportion the DC101 tumours exceed the control group and greatly exceed the Dll4 tumours, for which large vessels make up a negligible proportion of all vessels.

Interestingly, although other authors have found measures of tortuosity to be discriminative in imaging tumour vasculature at larger scales (Bullitt et al., 2003, 2005), we found neither the Chord-Length Ratio, or Sum-of-angles-metric to distinguish clearly between these three tumour cases. The results of this comparison are shown in Figure 7.7. Although the 'normalised' vasculature of the DC101 tumours errs on the side of less tortuosity, as would be expected (low SOAM, high CLR) these results do not appear significant enough to be discriminative and a larger sample size would be needed to further investigate this.

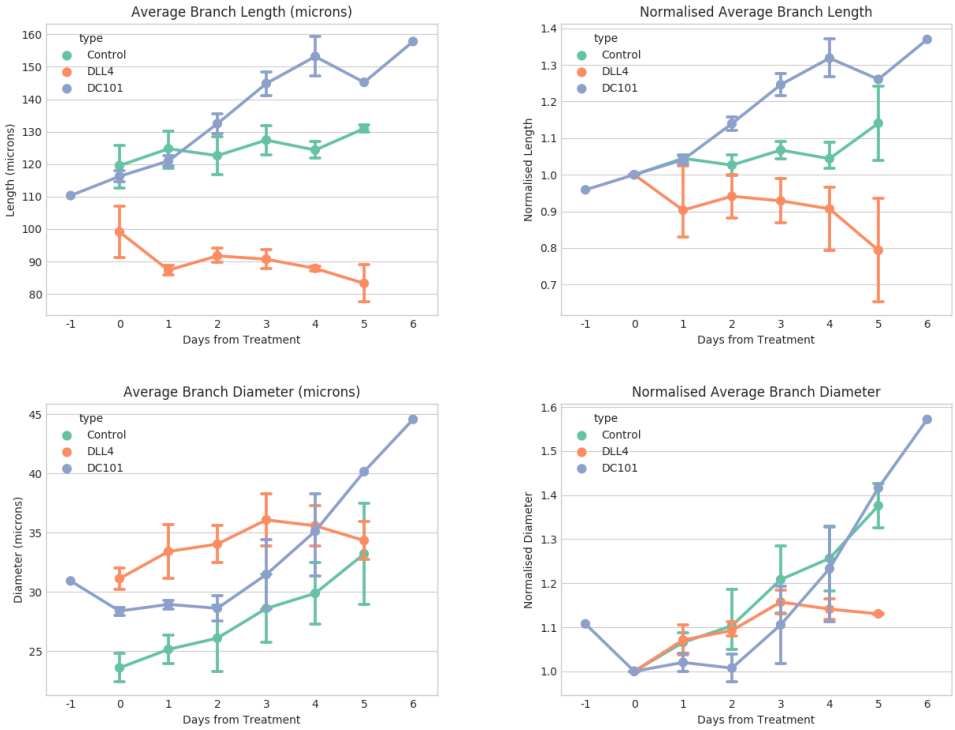


Figure 7.5: Changes in vessel length (top) and diameter (bottom) shown against days since treatment. Results are shown for the raw values (left) and the normalised values (right). Error bars show a 68% confidence interval for the mean.

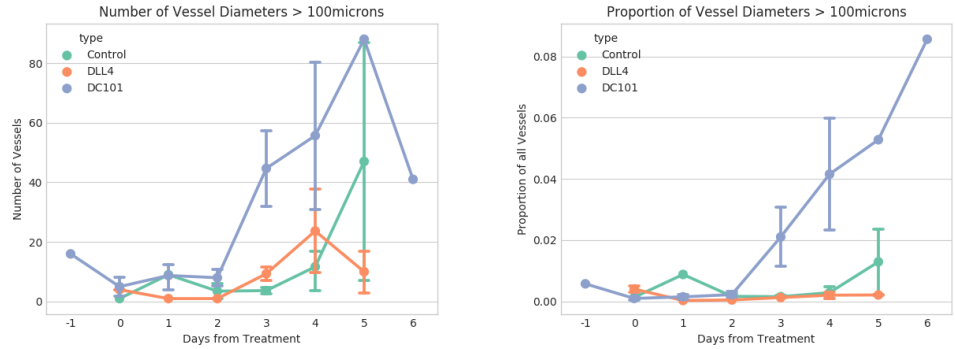


Figure 7.6: Left: Total number of vessels that exceed 100µm in diameter. Right: Proportion of vessels that exceed 100µm in diameter.

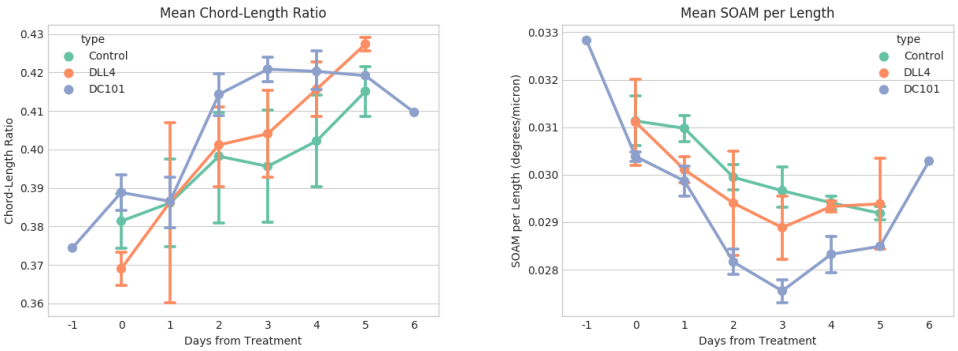


Figure 7.7: Comparison of the different tortuosity measurements Chord-Length Ratio (left) and Sum-of-angles-metric (right).

For the remainder of this chapter we will summarise and discuss the findings of this thesis and their significance to this application. We will also explore a number of avenues for further investigation in continuing this work.

7.2 Summary

In Chapter 3, we proposed and tested a method for segmentation, combining a novel supervoxel algorithm, a vessel orientated MRF and a learnt Random Forests model. We found this to be an effective way to segment the vascular network, specifically in a way which would be amenable to easy skeletonisation. We also note that this method is extremely fast to run, especially when compared to a previous state-of-the-art method for this application. This method is extremely well suited to a fast turn-around on novel experiments as it is able to be trained on very small data sets and the supervoxel formulation allows for extremely fast annotation, especially when combined with an active learning process to guide the annotator.

In Chapter 4, we began to explore the feasibility of using methods from deep learning for this data. We found that even with a fairly small training set we could achieve superior segmentation results by using a very large U-Net architecture, when compared a much more modest, flat architecture. We also explored the relationship between training via Dice based loss functions and cross-entropy based loss functions. We observed that while training with a Dice based coefficient may give an improved test Dice score, when compared to cross-entropy (after application of an optimal threshold), the Dice based output had discarded useful probabilistic information that may be leveraged as a measure of uncertainty. In this sense, the Dice coefficient based loss can be thought of as an over-fitting to the Dice evaluation metric and we therefore prefer a cross-entropy based loss. Going a step further than segmentation, we explored the possibility of encoding 3D information about the vessel centreline into a 2D model. In order to achieve this we proposed a transformation of the segmentation ground truth in order to approximate some 3D vesselness function. As an aside, we demonstrated that the calculation of a classical Hessian-based vesselness function is possible using fully differentiable CNN

machinery and suggest that this may be used as a surrogate or auxiliary loss. Finally, we evaluate the use of our vesselness proxy in extracting the skeletal structure of the vessel network. By doing this we find that by combining the vesselness and segmentation networks as an ensemble, we are able to achieve significantly improved results over the segmentation network alone.

In Chapter 5, we proposed the use of hybrid Convolutional-Recurrent networks as a lightweight approach to extending 2D CNNs into 3D in order to learn truly 3D skeletal structures. While in the previous chapter we attempted to encode the 3D skeleton in 2D, now we are able to learn the 3D skeleton directly and use the raw skeleton image as our training target. Although other authors have suggested large combined architectures of Convolutional-Recurrent layers, we found that the inclusion of a single ConvLSTM layer to a shared CNN architecture is sufficient to dramatically boost performance in this task. We tested both a deep and a shallow variant of this CNN+ConvLSTM hybrid and evaluated the use of both a weighted and regular cross-entropy loss function. We evaluated the component parts on synthetic data before comparing the optimal configuration to the method from the previous chapter. We observed that the inclusion of recurrent layers to allow 3D computations, significantly improves the ability to extract the vessel skeletons. We also observed that this improvement is most notable in complex and tightly packed vasculature where a 2D view alone is not sufficient to discriminate vessel from background.

In Chapter 6, we noted that although we were able to facilitate learning from skeletal targets using ConvLSTMs, it is not possible to learn structured outputs such as these using pixel-wise loss functions alone. We therefore proposed the inclusion of an adversarial loss, utilizing the conditional GAN framework in order to refine these centrelines. We motivate this with two illustrative examples, before demonstrating the improvements on our real data. We find that the inclusion of an adversarial loss reduces our dependence on tuning an optimal threshold. We also find that we are able to achieve this gain while retaining the power of the optimally thresholded method from Chapter 5.

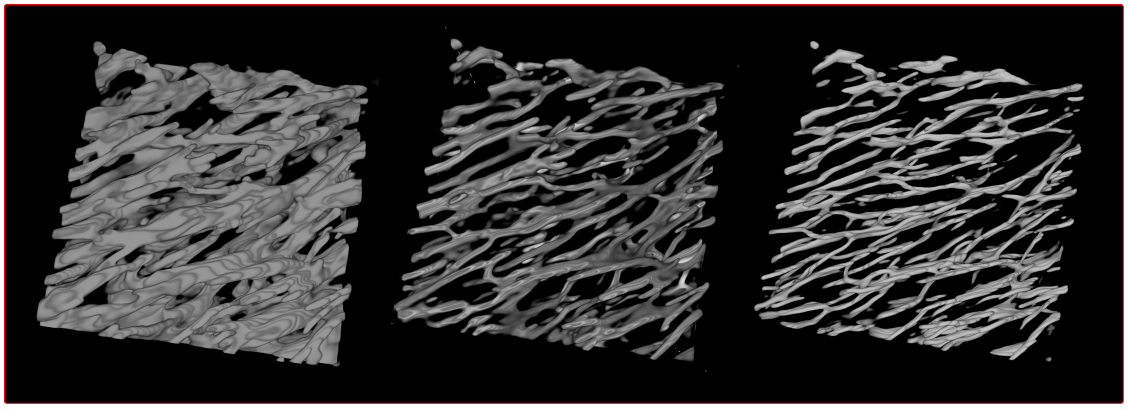


Figure 7.8: The progression of vessel extraction methods: Segmentation (left), Skeleton (centre) and Adversarial (right).

Finally, we demonstrate the application of these methods to real data examples. Specifically we use these methods to extract and analyse the vasculature from temporally imaged *in vivo* tumours and quantitatively comparing their details.

7.3 Discussion

The key observation of this thesis is that methods from deep learning provide a powerful set of tools for vessel segmentation, especially in complex and non-standard image modalities. We have seen that we are able to leverage the power of CNNs even without large public datasets and that a dataset created by a single researcher could feasibly be sufficient for even complex tasks like centreline extraction. Segmentation is one of the most common tasks in computer vision and appears to be the obvious first approach for extracting vasculature from our images. However, we observed that, in this case, a simple segmentation often obscures the true nature of the vascular network, especially for tightly packed and tortuous vasculature, such as that found in tumours. To this end, we found that by directing the algorithm to something closer to our final desired output (vessel centrelines) we were able to achieve an improved result. A demonstration of this improvement is shown in Figure 7.8. This is interesting as although the source of information has merely been transformed (all ground truth volumes are originally derived from a segmented volume) we are able to improve the utility of the model learned from it by injecting

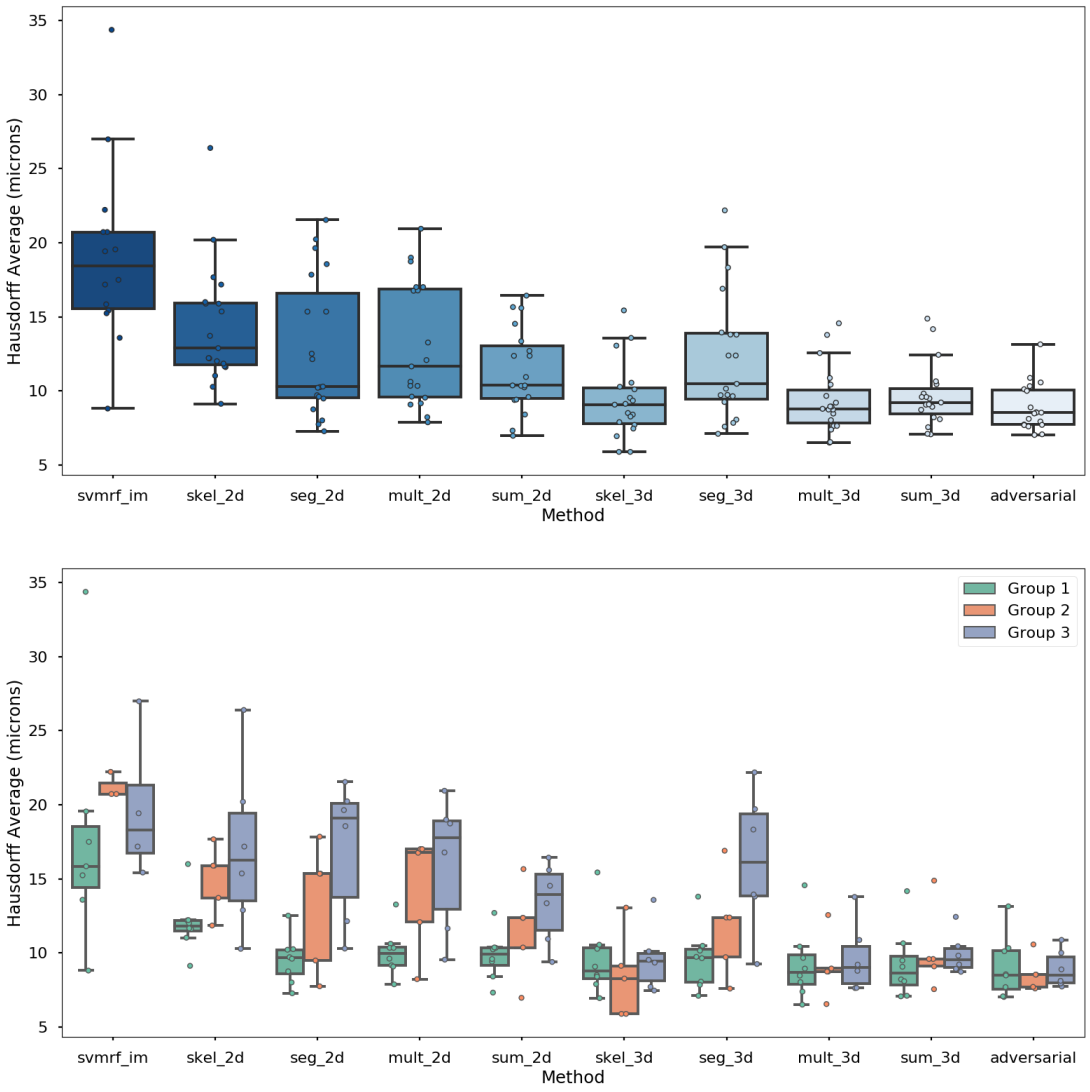


Figure 7.9: The progression of all methods proposed in the thesis (top) and the results for each method on each tumour category (right).

some prior knowledge about our final application. In this case our prior is extremely high level, although recent work on Bayesian Deep Learning is exploring the space of applying more general prior knowledge over the weights of the network directly. In Figure 7.9 we show the progression of all methods presented in this thesis.

Although we believe the methods in the later chapters provide a more powerful approach to the delineation of vasculature over the method presented in Chapter 3, there is still value in methods such as this. A key advantage is the ability to generate training sets in almost real time and perform active learning due to the

extremely fast training and execution of the random forests framework applied to supervoxels. Supervoxels also provide synergy in this by allowing us to annotate multiple pixels for training in a single brush-stroke as well as providing a useful overlay in terms of a kind of active-learning prompt.

We believe that Convolutional-Recurrent hybrid networks are an extremely powerful tool, that are well suited to a number of medical imaging modalities. Microscopy images that create z-stacks appear to be an ideal candidate for these approaches. Another benefit of this is that any existing 2D CNN network can be trivially extended to 3D by the minimal application of a single ConvLSTM layer. In Chapter 5 we observed that the shallow hybrid network consisted of only a 2.5% increase in the number of trainable weights and could be directly initialised using a pre-trained CNN, vastly increasing its inferential power for 3D structures.

The application of Adversarial training to learning skeletons reduced our dependence on finding an optimal threshold for the output of our network. However, this comes at the cost of a more opaque and complex training procedure. One consequence of this is that we have less clarity in terms of when to stop training, relying on a human-mediated early stopping procedure before pathological overfitting occurs. The most common failure modes of this kind of training were 'mechanical' ones, in which the discriminator would fail to keep up with the generator and thus cease to produce a useful gradient. Another common failure mode is overfitting in which the qualitative 'skeletal' property of the training set is learnt but the conditional nature is not, giving us a mass of single pixel wide skeletons which bare no resemblance to the input image. We do not view either of these as fatal flaws as these are very common issues which more recent methods aim to address. The Wasserstein GAN framework aims to address the issues of poorly defined discriminator gradients by replacing the discriminator loss function with an approximation to the Earth Mover's distance, which is well defined and non-zero everywhere. They also claim that in this framework the discriminator should be trained to convergence at each iteration, reducing the dependence on attempting to 'balance' the generator and discriminator, which can be problematic. We believe this to be a rich area for

future research as this appears to be a natural way to learn structures such as these, however, the method will require further innovation and optimisation.

The experiments in this thesis have lead us to the following conclusions:

1. Deep Convolutional Neural Networks are currently the most powerful tool available for semantic segmentation and related pixel-wise labelling tasks.

2. Although 2D CNNs can be effectively applied for the segmentation of volumetric data, they may be insufficient for more 3D oriented tasks such as vessel centreline extraction.

3. It is possible to construct a powerful 3D neural network by combining 2D CNNs with a single ConvLSTM unit, without adding a significant number of trainable weights.

4. Learning directly from centreline skeletons is an effective method for improving performance, beyond segmentation, for the delineation of tightly packed vasculature.

5. Weighted loss functions are vitally important for learning from such sparse structures. In addition to this, 3D CNNs (e.g. U-Net 3D) appear ineffective for this task.

6. Although we can learn an approximate version of these skeletons using supervised learning alone, an adversarial learning framework provides a way to genuinely train the network to produce centrelines directly.

7. Our adversarial refinement appears to removes the dependency on finding optimal thresholds. We observe results equivalent to the optimal results before refinement, across the majority of the range of possible thresholds.

8. Our synthetic experiments also suggest that it may help improve the topological integrity of the delineations.

7.4 Directions for Future Research

We believe that the work in this thesis opens up many avenues for further investigation and poses a number of questions that warrant attention. In this section we will suggest a number of directions that future work may take.

7.4.1 Adversarial Learning for Skeleton Extraction

As demonstrated in Chapter 6, adversarial learning can be a powerful tool for refining our models and allows us to compensate for subjective or qualitative annotations in training data. However, this emerging technique has a number of technical difficulties that we believe, with further research and refinement, could produce significantly improved results. The first area of investigation would be to explore the more recent developments in adversarial training schemes, such as WGAN (Arjovsky et al., 2017), Improved WGAN (Gulrajani et al., 2017) and InfoGAN (Chen et al., 2016d). This improved stability of training may lead to further improvements of the model directly, as improved minima in the loss landscape are reached. Secondly, further research into the architectural form of the discriminator network could yield improved results. In Chapter 6 we found that the DCGAN architecture, applied in the style of the PatchGAN was effective on both of our vessel tasks. However, the ImageGAN style was most effective on our toy shape example.

Although we motivated this method as a way to directly extract medial skeletons, we found that the more robust approach was to have the network learn very confident *almost*-skeletons and perform a trivial discrete skeletonisation step in order to reach the single pixel wide skeleton that we desired for analysis. However, we have observed that the networks are capable of learning this single pixel wide skeleton (as seen in Figure 6.7), but most often at the expense of robustness in the larger vessel topology. We believe that an improved method would be able to address both of these issues and entirely eliminate the skeletonisation process.

7.4.2 Unify into a Multi-Task Learning Framework

Currently our approach relies on training two networks, the skeletonisation network, focused on extracting the topology, and the segmentation network focused on segmenting the entire vasculature. Currently we use the segmentation network to derive the vessel diameter. However, this could potentially be performed by a third network, which would help improve measurements in tightly packed regions. When training networks on these different targets, there may be a synergy in sharing sections of the architecture between these three tasks. Multi-task learning presents a single network with a number of different objectives encoded in the loss function. This encourages the network to learn generically useful features as they must be applied to a diverse range of tasks. Different convergence rates, caused by a difference in the complexity of the tasks, can be addressed using a multi-task early stopping framework (Zhang et al., 2014). This has been shown to improve robustness but would certainly improve efficiency at the very least. Multi-task learning has been used to great effect already in natural language processing contexts (Seltzer and Droppo, 2013; Luong et al., 2016) as well as in face detection (Zhang et al., 2014) where it was shown to improve robustness to occlusion. In medical imaging, the effectiveness of multi-task learning has been demonstrated by Moeskops *et al.* (Moeskops et al., 2016), who attempted to train networks that were agnostic to imaging modality via a multi-task framework. We would be interested to see whether: firstly, a third network can be used to estimate vessel diameter rather than relying on distance maps and secondly whether these three networks can be mainly combined into a single network, trained using a multi-task loss.

7.4.3 Further Network Analysis

In Section 7.1 we presented a number of examples of the kind of analysis enabled by the methods proposed in this thesis. Although we presented a range of different and useful metrics such as vessel diameter, branching length, branch point density, vessel direction and various measurements of tortuosity, we believe that this analysis can go much further. A key component of all of these features is that they are not only

spatially distributed by they share a relationship with each other, described by the network topology of the vasculature. This network topology may provide further insights into the relationships within the growing tumour and the heterogeneity seen in the feature maps (see Figure 7.3).

In addition to this we also see the possibility of further longitudinal analysis. At present we are restricted in comparing average values between time points, but through the use of graph matching algorithms it may be possible to draw relationships between individual vessels or regions of the tumour, across time points. This would allow much richer trends to be extracted, especially in the context of the heterogeneity of features. As we saw in Figure 7.6, a significant trend was partially masked by the whole tumour averages. Being able to consider relationships between individual regions, or even specific vessels, between time points would provide much greater powers of investigation.

Finally, more and more mathematical models of tumour growth are utilising imaging data to provide realistic geometries, initial conditions and parameter calibration (Adhikarla and Jeraj, 2016; Grogan et al., 2017). We believe that this approach can provide a middle ground between pre-clinical imaging and mathematical modelling to extract extremely rich information from the images automatically and help create more accurate and predictive models.

Appendices

A

NetMets: Quantative Topological Measurements

For our topological measurements we use the various metrics proposed in the NetMets package by Mayerich *et al.* (Mayerich et al., 2012). These metrics are the Topological False-Positive Rate (T_{FPR}), Topological False-Negative Rate (T_{FNR}), Geometric False-Positive Rate (G_{FPR}) and Geometric False-Negative Rate (G_{FNR}) measures. We combine these to form following:

$$T_{max} = \min_t \max(T_{FPR}(t), T_{FNR}(t)) \tag{A.1}$$

and

$$G_{max} = \min_t \max(G_{FPR}(t), G_{FNR}(t)). \tag{A.2}$$

Here, $T_{FRP}(t)$ denotes the T_{FPR} score derived from an image thresholded at the value t and then skeletonised.

Intuitively, we can understand this value as a lower bound for the Area Under Curve (AUC) in a Receiver Operator Characteristic (ROC) analysis for situations where we can not trivially generate the entire curve. Since this curve is defined to be monotonic, the value of $1 - T_{max}$ gives the intersection between the ROC curve and the line $T_{FPR} = T_{FNR}$. This means that $(1 - T_{max})^2$ provides a lower bound for the AUC. We also see that $1 - T_{max}^2$ provides an upper bound for the AUC.

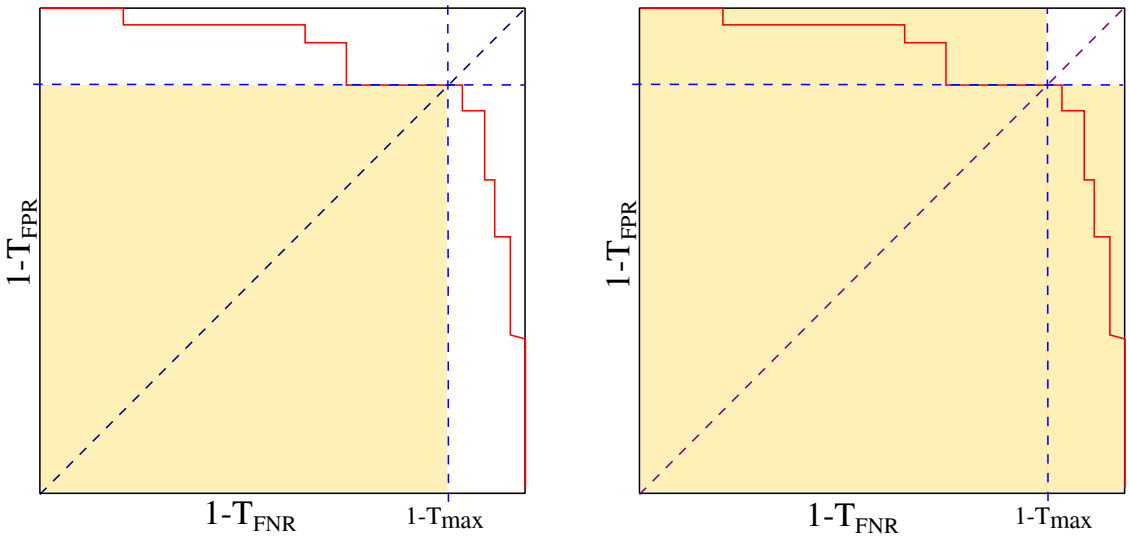


Figure A.1: A hypothetical ROC curve shown in red. Here we show how the values T_{FPR}, T_{FNR} and T_{max} all relate to an upper and lower bound on the AUC for the unknown ROC curve. Lower and upper bounds are shown in yellow on the left and right figures respectively.

This relationship is visualised in Figure A.1. We therefore claim that reducing T_{max} and G_{max} represents an improvement in the general ability of our vessel classifying networks that is distinct from the choice of threshold.

For the remainder of this section we will briefly outline the process by which we compute these topological agreement metrics, however, further details and discussion are provided in the work by Mayerich *et al.* (Mayerich et al., 2012). Although the Geometric measure is defined with an arbitrary metric in mind, we take it to measure the total length of the misaligned branches, as in the work by Almasi *et al.* (Almasi et al., 2015).

Overview: The core intuition behind these methods comes from the idea that two networks that attempt to represent the same physical structures will have some 'core' network that consists of the aspects of the physical structure that both networks agree upon. We describe each network as a graph, consisting of a set of vertices (or nodes) and a set of edges between those vertices.

$$G = \{V, E\}.$$

In this section we will assume that of the two networks to be compared, one is a ground-truth version of the underlying structure, GT, and one is our proposed network, T. We therefore attempt to form two 'core' networks: GT_c a sub-graph of GT, consisting of the aspects of GT that also exist in T and T_c defined similarly for T. Once GT_c and T_c have been constructed, our false-positive and false-negative rates can be directly calculated by comparison between T and T_c and between GT and GT_c. T_{FPR} is the ratio of edges that exist in T and not in T_c. T_{FNR} is the ratio of edges that exist in GT and not in GT_c. Similarly, G_{FPR} is the length of edges that exist in T and not in T_c divided by the total length of edges in T. G_{FNR} is the length of edges that exist in GT and not in GT_c divided by the total length of edges in GT. An example of these 'core' agreement graphs is shown in Figure A.2.

Core Graph Construction: In order to form the core graphs GT_c and T_c, we must first determine a correspondence between nodes in the graphs. This is done by a simple nearest neighbour search. Once a node has been assigned a corresponding node, it may not be assigned to any other. If no node exists within a radius of σ then no assignment is made. Once all possible assignments have been made we are left with a set of paired nodes:

$$V_c = \{(v_{1,GT}, v_{1,T}), \ldots, (v_{N,GT}, v_{N,T})\},$$

$$V_{c,GT} = \{v \mid v \in V_c \cap V_{GT}\},$$

$$V_{c,T} = \{v \mid v \in V_c \cap V_T\}.$$

We refer to these paired nodes as 'coloured' nodes.

Next we must determine the connectivity for these core graphs. In order to determine the edge set for the graph GT_c we consider each pair of nodes in $V_{c,GT}$. If GT contains an edge between the nodes $v_{i,GT}$ and $v_{j,GT}$ then we check the corresponding pair of nodes in T, namely, $v_{i,T}$ and $v_{j,T}$. Utilizing a shortest path algorithm (e.g. Dijkstra's algorithm), we find the shortest path between the nodes $v_{i,T}$ and $v_{j,T}$. If such a path exists, and it does not pass through any other coloured nodes (or any nodes used in previous paths), then we add an edge connecting

$v_{i,GT}$ and $v_{j,GT}$ to the edge set of GT_c. The edge set of T_c is determined similarly with the roles of GT and T reversed.

Metric Computation: Once we have constructed the core graphs T_c and GT_c, the metrics that we will use are computed as follows:

- T_{FPR}: Count the number of uncoloured nodes in T as well as the number of edges in T that are not used to construct T_c. Divide this by the total number of edges in T.
- T_{FNR}: Count the number of uncoloured nodes in GT as well as the number of edges in GT that are not used to construct GT_c. Divide this by the total number of edges in GT.
- G_{FPR}: Sum the total length edges in T that are not used to construct T_c. Divide this by the total length of all edges in T.
- G_{FNR}: Sum the total length edges in GT that are not used to construct GT_c. Divide this by the total length of all edges in GT.
- T_{max}: The maximum value of T_{FPR} and T_{FNR}.
- G_{max}: The maximum value of G_{FPR} and G_{FNR}.

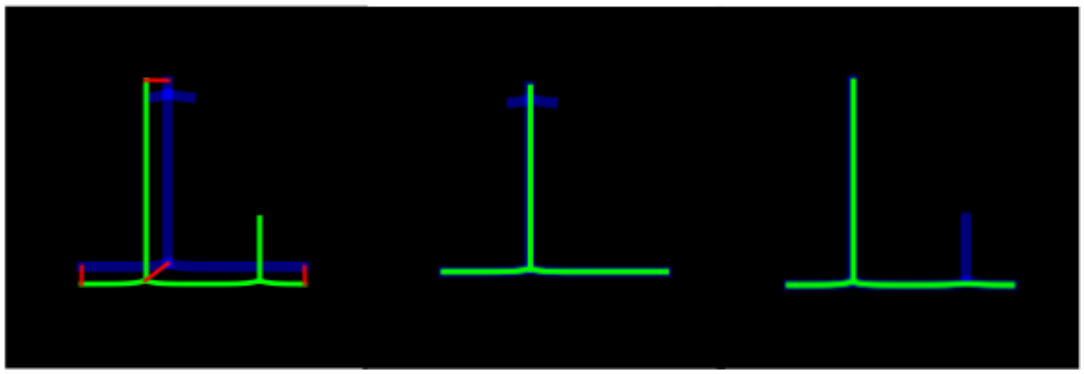


Figure A.2: Example of the various graph constructions. Left: GT graph (blue) and T graph (green) with node assignment shown (red). Centre: GT graph (blue) is overlaid with the core connectivity graph GT_c (green). Right: T graph (blue) is overlaid with the core connectivity graph T_c (green).

B
Statistical Testing

In this appendix, we will detail the statistical tests and techniques used throughout this thesis. As our tests largely consist of comparisons between multiple methods being applied to the same data points we consider a paired significance test, the Wilcoxon Signed-Rank Test, and a correction scheme for multiple comparisons, Holm-Bonferroni correction.

B.1 Wilcoxon Signed-Rank Test

The Wilcoxon Signed-Rank test is a non-parametric test used to test for significance between two sets of paired samples. This means that, when testing for significant differences between set A and set B, for each point in set A there exists a corresponding point in set B. For our purposes this correspondence comes from the fact that each model is tested on the same test images, so we have an entry from each image in each set, which can be matched. Unlike the paired Student's t-test, the Wilcoxon Signed-Rank test does not assume normality of the samples.

Considering two paired populations $x_1, \ldots, x_N$ and $y_1, \ldots, y_N$, we consider the null hypothesis H_0 and the alternative hypothesis H_1:

- H_0: difference between the pairs follows a symmetric distribution around zero

- H_1: difference between the pairs does not follow a symmetric distribution around zero.

The test procedure is as follows, considering our :

1. Compute $|x_i - y_i|$ and $sign(x_i - y_i)$, discarding pairs for which $|x_i - y_i| = 0$.
2. Order the remaining pairs from smallest to largest absolute difference.
3. Assign a rank R_i to each pair according to this order, beginning with 1.
4. Compute test statistic $W = \sum_{i=1}^{N}[\text{sgn}(x_i - y_i) \cdot R_i]$, where N is the number of ranked pairs.
5. Compare $|W|$ to critical values in a reference table for statistical significance at a given p-value.

B.2 Holm-Bonferroni Correction

The Holm-Bonferroni procedure aims to control the family-wise error rate when performing multiple comparisons on a single dataset. Performing multiple comparisons increases the likelihood of Type I errors (false-positives). We should therefore control our significance level such that we do not influence the rate of false positives by performing more than one comparison. The Bonferroni correction procedure addresses this by modifying the significance level α for all comparisons such that

$$\hat{\alpha} = \frac{\alpha}{N}, \tag{B.1}$$

for N comparisons. However, this may be considered an over-correction and may introduce unnecessary Type II errors (false-negative) compared to other methods. The Holm-Bonferroni procedure is uniformly more powerful than the Bonferroni procedure alone and is performed as follows, for a significance level α:

1. Perform N hypothesis tests and acquire a list of p-values, $\{p_1, \ldots, p_n\}$.
2. Order these p-values from lowest to highest such that $\hat{p}_1 < \hat{p}_2 < \cdots < \hat{p}_N$.

3. Assign the rank k to p-value, p_i, which corresponds to the ranked p-value $\hat{p}_k$.

4. Test significance for each p-value p_i^k at the level

$$\hat{\alpha_k} = \frac{\alpha}{N - k + 1}.$$

C

Skeletonisation Loss Functions

In addition to learning representations that maximise some kind of vessel-like morphology it may be possible force the network to go one step further and learn to output some topological skeleton. We can do this by tailoring a loss to penalise non-skeletal outputs. The skeleton is defined as a binary image, where, with the exception of branching points and end points, each pixel has exactly 2 non-zero neighbours. We can reach such a skeleton by sequentially removing pixels until no more can be removed without altering the topology of the binary image. Alternately, we would like to train neural networks that are able to generate such skeletons directly. With the network able to take a more holistic view of the image we may eliminate a lot of the artefactual branches left by skeletonisation via local thinning, which is an ill-posed problem with many valid skeletons for each binary image.

Here we propose two potential implementations of skeletonisation via CNN.

Neighbourhood The number of neighbours in a binary image for each pixel can be calculated directly via convolution with a kernel K of the form

$$K_4 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{C.1}$$

or

$$K_8 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{C.2}$$

depending on the kind of neighbourhood we consider.

We can then construct a loss function that attempts to maximise skeletal connectivity on the foreground regions and minimise the number of neighbours elsewhere. Where y_i indicates the value of some binary target at position i and n_i indicates the number of neighbours, computed by convolution with kernel K, detected at position i.

$$\mathcal{L} = \sum_i y_i(n_i - 2)^2 + n_i(1 - y_i) \tag{C.3}$$

However, although this formulation is valid for the branch sections of a skeleton, it will become problematic at end points, where the number of neighbours will be 1, and at junctions, where the number of neighbours will be greater than two. This may have the combined effects of continually shortening branches, and failing to create junction points. A further difficulty with methods such as this are that in our neighbour calculus we assume a binary image. In order to create a binary image we are required to perform some kind of thresholding operation, however such an operation is non-differentiable and therefore cannot be used without our network. Methods of inference on discrete variables using smooth functions has been proposed jointly by Jang *et al.* with their Gumbel Softmax distribution (Jang et al., 2017) and by Maddison *et al.* (Maddison et al., 2016) with their Concrete Distribution. Both methods use the 're-parametrisation trick' to approximate sampling from a parametrised categorical distribution in a fully differentiable way.

Template Matching An alternate method of skeletonisation is to consider whether each pixel is integral to the local topology of the image, and if not then remove it. In the case of 2D thinning the following two templates, along with their

rotations will suffice (Jain, 1989; Gonzalez, 1977):

$$T_1 = \begin{pmatrix} 0 & 0 & 0 \\ & 1 & \\ 1 & 1 & 1 \end{pmatrix} \tag{C.4}$$

$$T_2 = \begin{pmatrix} & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & \end{pmatrix} \tag{C.5}$$

For each 3-by-3 neighbourhood in our binary image, if we match one of these templates (blank spaces may take any value) then we may safely remove the central pixel from the image without affecting the topology. This procedure may be applied iteratively and will converge on a skeletal representation of the binary image.

We can approximate this approach using convolutions in the following way. Consider a binary image I of shape [N,M]. We first concatenate this image with its own negation, $1 - I$ in a new axis to create a new image tensor of shape [N,M,2]. We then split the templates over both images in such a way that they characterise the disagreement between the image and the template. For example the template T_1 from Equation C.4, becomes:

$$\bar{T}_{1,1} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{C.6}$$

$$\bar{T}_{1,2} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \tag{C.7}$$

We then take the minimum response and transform it with a sigmoid non-linearity.

$$F = 1 - \sigma\left(\min_i \left(I * \bar{T}_i\right)\right) \tag{C.8}$$

By minimizing F we maximise the minimal disagreement and, by doing this, aim to remove all pixels from the binary image that are not integral to a skeletal representation. This loss should be minimised when all matching regions have been removed from the image leaving a skeletal representation.

Bibliography

R. Achanta, A. Shaji, K. Smith, G. Knott, and P. Fua. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

V. Adhikarla and R. Jeraj. An imaging-based computational model for simulating angiogenesis and tumour oxygenation dynamics. *Physics in Medicine and Biology*, 61(10):3885–3902, may 2016.

S. Almasi, X. Xu, A. Ben-Zvi, B. Lacoste, C. Gu, and E. L. Miller. A novel method for identifying a graph-based representation of 3-D microvascular networks from fluorescence microscopy image stacks. *Medical Image Analysis*, 20:208–223, 2015.

S. Almasi, A. Ben-Zvi, B. Lacoste, C. Gu, E. L. Miller, and X. Xu. Joint volumetric extraction and enhancement of vasculature from low-SNR 3-D fluorescence microscopy images. *Pattern Recognition*, 63:710–718, 2016.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, page 30, 2017.

Y. M. Assael, B. Shillingford, S. Whiteson, and N. De Freitas. LipNet: End-to-end Sentence-level Lipreading. *arXiv preprint arXiv:1611.01599*, 2016.

V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv preprint arXiv:1511.00561*, 2016.

P. R. Barber, B. Vojnovic, S. M. Ameer-Beg, R. J. Hodgkiss, G. M. Tozer, and J. Wilson. Semi-automated software for the three-dimensional delineation of complex vascular networks. *Journal of Microscopy*, 211(Pt 1):54–62, jul 2003.

S. Basu, B. Condron, and S. T. Acton. Path2Path: Hierarchical path-based analysis for neuron matching. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 996–999. IEEE, mar 2011.

R. Bates, L. Risser, B. Irving, B. Papiez, P. Kannan, V. Kersemans, and J. Schnabel. Filling Large Discontinuities in 3D Vascular Networks using Skeleton- and Intensity-based Information. In *Proc. MICCAI*. Springer Berlin Heidelberg, 2015.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

K. Bentley, C. A. Franco, A. Philippides, R. Blanco, M. Dierkes, V. Gebala, F. Stanchi, M. Jones, I. M. Aspalter, G. Cagna, S. Weström, L. Claesson-Welsh, D. Vestweber, and H. Gerhardt. The role of differential VE-cadherin dynamics in cell rearrangement during angiogenesis. *Nature Cell Biology*, 16(4):309–321, 2014.

G. Bergers and D. Hanahan. Modes of resistance to anti-angiogenic therapy. *Nature Reviews Cancer*, 8(8):592–608, 2008.

S. Bériault, M. Archambault-Wallenburg, A. F. Sadikot, D. Louis Collins, and G. Bruce Pike. Automatic Markov Random Field Segmentation of Susceptibility-Weighted MR Venography. In *Clinical Image-Based Procedures. Translational Research in Medical Imaging*, pages 39–47. Springer, 2014.

J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 192–236, 1974.

Y. Boykov and G. Funka-Lea. Graph Cuts and Efficient ND Image Segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (11):1222–1239, 2001.

Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in ND images. In *Proc. ICCV*, pages 105–112, 2001.

L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

E. Bullitt, G. Gerig, S. Pizer, W. Lin, and S. Aylward. Measuring Tortuosity of the Intracerebral Vasculature From MRA Images. *IEEE Transactions on Medical Imaging*, 22(9):1163–1171, 2003.

E. Bullitt, D. Zeng, G. Gerig, S. Aylward, S. Joshi, J. Smith, W. Lin, and M. Ewend. Vessel Tortuosity and Brain Tumor Malignancy : a blinded study. *Academic radiology*, 12(10):1232–1240, 2005.

W. Chan, N. Jaitly, Q. V. Le, O. Vinyals, and G. Brain. Listen, Attend and Spell. *arXiv preprint arXiv:1508.01211*, 2015.

C. Chen, A. Liaw, and L. Breiman. Using Random Forest to Learn Imbalanced Data. 2004.

H. Chen, Q. Dou, L. Yu, and P.-A. Heng. VoxResNet: Deep Voxelwise Residual Networks for Volumetric Brain Segmentation. *arXiv preprint arXiv:1608.05895*, 2016a.

J. Chen, L. Yang, Y. Zhang, M. Alber, and D. Z. Chen. Combining Fully Convolutional and Recurrent Neural Networks for 3D Biomedical Image Segmentation. *Proc. NIPS*, 2016b.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv preprint arXiv:1606.00915*, 2016c.

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Proc. NIPS*, pages 2172–2180, 2016d.

J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip Reading Sentences in the Wild. *arXiv preprint arXiv:1611.05358*, 2016.

Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3D U-net: Learning dense volumetric segmentation from sparse annotation. In *Proc. MICCAI*, volume 9901 LNCS, pages 424–432, jun 2016.

A. Coates and A. Y. Ng. Learning Feature Representations with K-Means. In *Neural Networks: Tricks of the Trade*, pages 561–580. Springer Berlin Heidelberg, 2012.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (5):603–619, 2002.

J. P. B. O. Connor, C. J. Rose, A. Jackson, Y. Watson, S. Cheung, F. Maders, B. J. Whitcher, C. Roberts, G. A. Buonaccorsi, G. Thompson, A. R. Clamp, G. C. Jayson, and G. J. M. Parker. DCE-MRI biomarkers of tumour heterogeneity predict CRC liver metastasis shrinkage following bevacizumab and FOLFOX-6. *British Journal of Cancer Research*, 105(June):139–145, 2011.

A. Criminisi, J. Shotton, and E. Konukoglu. Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3):81–227, 2011.

F. Danhier. To exploit the tumor microenvironment: Since the EPR effect fails in the clinic, what is the future of nanomedicine? *Journal of Controlled Release*, 244:108–121, 2016.

E. Denton, S. Gross, and R. Fergus. Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks. *arXiv preprint arXiv:1611.06430*, nov 2016.

P. Dollár, Z. Tu, and S. Belongie. Supervised Learning of Edges and Object Boundaries. In *Proc. CVPR*, 2006.

M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal. The Importance of Skip Connections in Biomedical Image Segmentation. In *International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis.*, pages 179–187. Springer International Publishing, aug 2016.

M.-P. Dubuisson and A. Jain. A modified Hausdorff distance for object matching. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 566–568. IEEE Comput. Soc. Press, 1994.

J. Ehling, B. Theek, and F. Gremse. Micro-CT imaging of tumor angiogenesis: quantitative measures describing micromorphology and vascularization. *The American journal of pathology*, 184(2):431–441, 2014.

D. Engel, L. Spinello, R. Triebel, R. Siegwart, H. Bulthoff, and C. Curio. Medial Features for Superpixel Segmentation. In *Proc. MVA*, pages 248–252, 2009.

T. Fang, R. Su, L. Xie, Q. Gu, Q. Li, P. Liang, and T. Wang. Retinal vessel landmark detection using deep learning and hessian matrix. In *2015 8th International Congress on Image and Signal Processing (CISP)*, pages 387–392. IEEE, oct 2015.

P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

L. Feng, T. Zhao, and J. Kim. neuTube 1.0: a New Design for Efficient Neuron Reconstruction Software Based on the SWC Format. *eNeuro*, 2(1):1–10, 2015.

M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, and D. Amorim Fernández-Delgado. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014.

L. Fidon, W. Li, L. C. Garcia-Peraza-Herrera, J. Ekanayake, N. Kitchen, S. Ourselin, and T. Vercauteren. Generalised Wasserstein Dice Score for Imbalanced Multi-class Segmentation using Holistic Convolutional Networks. *arXiv preprint arXiv:1707.00478*, jul 2017.

A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale Vessel Enhancement Filtering. In *Proc. MICCAI*, pages 130–137. Springer Berlin Heidelberg, 1998.

W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:891–906, 1991.

K. Fukushima. Biological Cybernetics Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biol. Cybernetics*, 36(193), 1980.

K. Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition - ScienceDirect. *Neural Networks*, 1(2):199–130, 1988.

B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. *Proc. ICCV*, 9:670–677, 2009.

S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, nov 1984.

T. A. Gillette, K. M. Brown, and G. A. Ascoli. The DIADEM Metric: Comparing Multiple Reconstructions of the Same Neuron. *Neuroinformatics*, 9(2-3):233–245, sep 2011.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 9:249–256, 2010.

R. Gonzalez. *Digital image processing.* 1977.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Proc. NIPS*, pages 2672–2680, 2014.

A. Gooya, H. Liao, and I. Sakuma. Generalization of geometrical flux maximizing flow on Riemannian manifolds for improved volumetric blood vessel segmentation. *Computerized Medical Imaging and Graphics*, 36(6):474–83, sep 2012.

D. J. Gould, T. J. Vadakkan, R. A. Poché, and M. E. Dickinson. Multifractal and Lacunarity Analysis of Microvascular Morphology and Remodeling. *Microcirculation*, 18(2), 2011.

V. Grau, J. C. Downs, and C. Burgoyne. Segmentation of trabeculated structures using an anisotropic Markov random field: application to the study of the optic nerve head in glaucoma. *IEEE Transactions on Medical Imaging*, 25(3):245–55, mar 2006.

A. Graves. Generating Sequences With Recurrent Neural Networks. *arXiv preprint arXiv:1308.0850*, aug 2013.

A. Graves and J. Schmidhuber. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. In *Proc. NIPS*, pages 545–552, 2009.

A. Graves, N. Jaitly, and A.-R. Mohamed. Hybrid Speech Recognition With Deep Bidirectional LSTM. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013a.

A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. IEEE, may 2013b.

J. A. Grogan, B. Markelc, A. J. Connor, R. J. Muschel, J. M. Pitt-Francis, P. K. Maini, and H. M. Byrne. Predicting The Influence of Microvascular Structure On Tumour Response to Radiotherapy. *IEEE Transactions on Biomedical Engineering*, 64(3):504 – 511, 2017.

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved Training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*, mar 2017.

D. Hanahan and R. A. Weinberg. The Hallmarks of Cancer. *Cell*, 100:57–70, 2000.

D. Hanahan and R. A. Weinberg. Hallmarks of cancer: the next generation. *Cell*, 144(5):646–74, mar 2011.

K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. CVPR*, pages 770–778, 2016.

M. Heinrich, M. Jenkinson, B. Papiez, F. Gleeson, M. Brady, and J. Schnabel. Edge- and detail-preserving sparse image representations for deformable registration of chest MRI and CT volumes. *Information Processing in Medical Imaging.*, 20(9): 463–474, 2013.

H. Heumann and G. Wittum. The Tree-Edit-Distance, a Measure for Quantifying Neuronal Morphology. *Neuroinformatics*, 7(3):179–190, sep 2009.

G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015.

T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

M. Holzer and R. Donner. Over-Segmentation of 3D Medical Image Volumes based on Monogenic Cues. In *Proc. CVWW*, pages 35–42, 2014.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. *arXiv preprint arXiv:1608.06993*, 2016.

J.-Y. Huang, L.-T. Li, H. Wang, S.-S. Liu, Y.-M. Lu, M.-H. Liao, R.-R. Tao, L.-J. Hong, K. Fukunaga, Z. Chen, C. Wilcox, E. Y. Lai, and F. Han. In vivo two-photon fluorescence microscopy reveals disturbed cerebral capillary blood flow and increased susceptibility to ischemic insults in diabetic mice. *CNS Neuroscience & Therapeutics*, 20(9):816–22, 2014.

D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, pages 448–456, feb 2015.

B. Irving, A. Cifor, B. Papiez, J. Franklin, E. Anderson, M. Brady, and J. Schnabel. Automated Colorectal Tumour Segmentation in DCE-MRI Using Supervoxel Neighbourhood Contrast Characteristics. In *Proc. MICCAI*, pages 609–616, 2014a.

B. Irving, A. Cifor, B. W. Papież, J. Franklin, E. M. Anderson, S. M. Brady, and J. A. Schnabel. Automated Colorectal Tumour Segmentation in DCE-MRI Using Supervoxel Neighbourhood Contrast Characteristics. In *Proc MICCAI*, pages 609–616. Springer International Publishing, 2014b.

B. Irving, J. M. Franklin, B. W. Papież, E. M. Anderson, R. A. Sharma, F. V. Gleeson, S. M. Brady, and J. A. Schnabel. Pieces-of-parts for supervoxel segmentation with global context: Application to DCE-MRI tumour delineation. *Medical Image Analysis*, 32:69–83, 2016.

P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv preprint arXiv:1611.07004*, nov 2016.

A. Jackson, J. P. B. O. Connor, and G. J. M. Parker. Imaging Tumor Vascular Heterogeneity and Angiogenesis using Dynamic Contrast-Enhanced Magnetic Resonance Imaging. *Clinical Cancer Research*, 13(12):3449–3459, 2007.

A. Jain. *Fundamentals of digital image processing*. 1989.

R. Jain. Normalizing tumor vasculature with anti-angiogenic therapy: a new paradigm for combination therapy. *Nature Medicine*, 7(9):987–989, 2001.

E. Jang, S. Gu, and B. Poole. Categorical Reparameterization with Gumbel-Softmax. *arXiv preprint arXiv:1611.01144*, 2017.

S. Jegou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1175–1183. IEEE, jul 2017.

N. Kalchbrenner, I. Danihelka, and A. Graves. Grid Long Short-Term Memory. *arXiv preprint arXiv:1507.01526*, jul 2015.

K. Kamnitsas, C. Baumgartner, C. Ledig, V. F. J. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, A. Nori, A. Criminisi, D. Rueckert, and B. Glocker. Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. *International Conference on Information Processing in Medical Imaging*, dec 2016a.

K. Kamnitsas, C. Ledig, V. F. J. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61–78, 2016b.

V. Kaynig, T. Fuchs, and J. M. Buhmann. Neuron geometry extraction by perceptual grouping in ssTEM images. In *Proc. CVPR*, pages 2902–2909, 2010.

V. Kaynig, A. Vazquez-Reina, S. Knowles-Barley, M. Roberts, T. R. Jones, N. Kasthuri, E. Miller, J. Lichtman, and H. Pfister. Large-Scale Automatic Reconstruction of Neuronal Processes from Electron Microscopy Images. *Medical Image Analysis*, 2015.

V. Kersemans, P. Kannan, J. S. Beech, R. Bates, B. Irving, S. Gilchrist, P. D. Allen, J. Thompson, P. Kinchesh, C. Casteleyn, J. Schnabel, M. Partridge, R. J. Muschel, and S. C. Smart. Improving In Vivo High-Resolution CT Imaging of the Tumour Vasculature in Xenograft Mouse Models through Reduction of Motion and Bone-Streak Artefacts. *PloS one*, 10(6):e0128537, jan 2015.

A. Kessy, A. Lewin, and K. Strimmer. Optimal whitening and decorrelation. *The American Statistician*, dec 2017.

J. S. Kim, M. J. Greene, A. Zlateski, K. Lee, M. Richardson, S. C. Turaga, M. Purcaro, M. Balkam, A. Robinson, B. F. Behabadi, M. Campos, W. Denk, and H. S. Seung. Space-time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500):331–6, may 2014.

D. P. Kingma and J. L. Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations 2015*, pages 1–15, dec 2015.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, dec 2013.

C. Kirbas and F. Quek. Vessel extraction techniques and algorithms: a survey. In *Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering*, 2003.

R. Kiros, K. Popuri, D. Cobzas, and M. Jagersand. Stacked Multiscale Feature Learning for Domain Independent Medical Image Segmentation. 2012.

P. Kohli, L. Ladický, and P. H. S. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. *International Journal of Computer Vision*, 82(3): 302–324, jan 2009.

V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *Proc. ICCV*, volume 1, pages 564–571, 2005.

V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2): 147–59, feb 2004.

P. Krähenbühl and V. Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *Proc. NIPS*, pages 109–117, 2012.

K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset. Model based multiscale detection of 3D vessels. In *IEEE Workshop on Biomedical Image Analysis*, pages 202–210. IEEE Comput. Soc, 1998.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proc. NIPS*, pages 1097–1105, 2012.

A. Lahiri, K. Ayush, P. K. Biswas, and P. Mitra. Generative Adversarial Learning for Reducing Manual Annotation in Semantic Segmentation on Large Scale Miscroscopy Images: Automated Vessel Segmentation in Retinal Fundus Image as Test Case. In *Proc. CVPR*, pages 794–800. IEEE, jul 2017.

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

K. Lee, A. Zlateski, A. Vishwanathan, and H. S. Seung. Recursive Training of 2D-3D Convolutional Networks for Neuronal Boundary Detection. In *Proc. NIPS*, pages 3573–3581, 2015.

D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea. A review of 3D vessel lumen segmentation techniques: models, features and extraction schemes. *Medical Image Analysis*, 13(6):819–45, dec 2009.

Q. Li, B. Feng, L. Xie, P. Liang, H. Zhang, and T. Wang. A Cross-Modality Learning Approach for Vessel Segmentation in Retinal Images. *IEEE Transactions on Medical Imaging*, 35(1):109–118, jan 2016.

R. Li, W. Zhang, H.-I. Suk, L. Wang, J. Li, D. Shen, and S. Ji. Deep Learning Based Imaging Data Completion for Improved Brain Disease Diagnosis. In *Proc. MICCAI*, pages 305–312. Springer, Cham, 2014.

P. Liskowski and K. Krawiec. Segmenting Retinal Blood Vessels With Deep Neural Networks. *IEEE Transactions on Medical Imaging*, 35(11):2369–2380, nov 2016.

G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez. A Survey on Deep Learning in Medical Image Analysis. *arXiv preprint arXiv:1702.05747*, feb 2017.

J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proc. CVPR*, pages 3431–3440, 2015.

C. Lorenz, I. Carlsen, and T. Buzug. Multi-scale line segmentation with automatic estimation of width, contrast and tangential direction in 2D and 3D medical images. In *In CVRMed-MRCAS'97*, pages 233–242. Springer Berlin/Heidelberg, 1997.

P. Luc, C. Couprie, S. Chintala, and J. Verbeek. Semantic Segmentation using Adversarial Networks. *arXiv preprint arXiv:1611.08408*, nov 2016.

A. Lucchi, K. Smith, R. Achanta, G. Knott, and P. Fua. Supervoxel-based segmentation of mitochondria in EM image stacks with learned shape features. *IEEE, Transactions on Medical Imaging*, 34(11):2274–2282, 2012.

M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, L. Kaiser, and G. Brain. Multi-task Sequence to Sequence Learning. In *Proc. ICLR*, 2016.

M. Lyksborg, O. Puonti, M. Agn, and R. Larsen. An Ensemble of 2D Convolutional Neural Networks for Tumor Segmentation. In *Scandinavian Conference on Image Analysis*, pages 201–211. Springer, Cham, 2015.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297. The Regents of the University of California, 1967.

C. J. Maddison, A. Mnih, and Y. W. Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *arXiv preprint arXiv:1611.00712*, 2016.

R. Manniesing, M. Viergever, and W. Niessen. Vessel Enhancing Diffusion. *Medical Image Analysis*, 10(6):815–825, 2006a.

R. Manniesing, M. A. Viergever, and W. J. Niessen. Vessel enhancing diffusion: A scale space representation of vessel structures. *Medical Image Analysis*, 10(6): 815–825, 2006b.

D. Mayerich, C. Bjornsson, J. Taylor, and B. Roysam. NetMets: software for quantifying and visualizing errors in biological network segmentation. *BMC Bioinformatics*, 13 Suppl 8(Suppl 8):S7, 2012.

E. F. Meijer, J. W. Baish, T. P. Padera, and D. Fukumura. Measuring Vascular Permeability In Vivo. *Methods Mol Biol*, 1458:71–85, 2016.

F. Milletari, N. Navab, and S.-A. Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, oct 2016.

M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784*, nov 2014.

J. Missbach-Guentner, J. Hunia, and F. Alves. Tumor blood vessel visualization. *The International Journal of Developmental Biology*, 55(4-5):535–46, jan 2011.

P. Moeskops, J. M. Wolterink, B. H. M. van der Velden, K. G. A. Gilhuijs, T. Leiner, M. A. Viergever, and I. Išgum. Deep Learning for Multi-task Medical Image Segmentation in Multiple Modalities. In *Proc. MICCAI*, pages 478–486. Springer, Cham, oct 2016.

A. Mottini, X. Descombes, and F. Besse. From Curves to Trees: A Tree-like Shapes Distance Using the Elastic Shape Analysis Framework. *Neuroinformatics*, 13: 175–191, 2015.

A. Narayanaswamy, S. Dwarakapuram, C. Bjornsson, B. Cutler, W. Shain, and B. Roysam. Robust Adaptive 3-D Segmentation of Vessel Laminae From Fluorescence Confocal Microscope Images and Parallel GPU Implementation. *IEEE Transactions on Medical Imaging*, 29(3):583–597, mar 2010.

H. Oda, K. Bhatia, M. Oda, T. Kitasaka, S. Iwano, H. Homma, H. Takabatake, M. Mori, H. Natori, J. A. Schnabel, and K. Mori. Hessian-assisted supervoxel: structure-oriented voxel clustering and application to mediastinal lymph node detection from CT volumes. In *Proc. SPIE, Medical Imaging: Computer-Aided Diagnosis*, page 101341D, mar 2017.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. *Proceedings of The 30th International Conference on Machine Learning*, (2):1310–1318, 2012.

D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context Encoders: Feature Learning by Inpainting. In *Proc. CVPR*, pages 2536–2544, 2016.

H. Peng, M. Hawrylycz, J. Roskams, S. Hill, N. Spruston, E. Meijering, and G. A. Ascoli. BigNeuron: Large-Scale 3D Neuron Reconstruction from Optical Microscopy Images. *Neuron*, 87(2):252–256, jul 2015.

P. Pinheiro and R. Collobert. Recurrent Convolutional Neural Networks for Scene Labeling. In *Proc. PMLR*, pages 82–90, jan 2014.

M. J. Pittet, R. Weissleder, E. Lin, Y. Wang, F. Pixley, E. Stanley, T. Graf, J. Pollard, J. Segall, J. Condeelis, and et Al. Intravital imaging. *Cell*, 147(5): 983–91, nov 2011.

S. Pizer, C. Burbeck, and J. Coggins. Object shape before boundary shape: Scale-space medial axes. *Journal of Mathematical Imaging and Vision*, 4(3):303–313, 1994.

R. P. Poudel, P. Lamata, and G. Montana. Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation. In *International Workshop on Reconstruction and Analysis of Moving Body Organs.*, volume 10129 LNCS, pages 83–94, aug 2017.

A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen. Deep Feature Learning for Knee Cartilage Segmentation Using a Triplanar Convolutional Neural Network. In *Proc. MICCAI*, pages 246–253. Springer, Berlin, Heidelberg, 2013.

P. Prentašic, M. Heisler, Z. Mammo, S. Lee, A. Merkur, E. Navajas, M. F. Beg, M. Šarunic, and S. Loncaric. Segmentation of the foveal microvasculature using deep learning networks. *Journal of Biomedical Optics*, 21(7):075008, jul 2016.

C. Pudney. Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images. *Computer Vision and Image Understanding*, 72(3):404–413, dec 1998.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2015.

L. Ritsma, E. J. A. Steller, S. I. J. Ellenbroek, O. Kranenburg, I. H. M. Borel Rinkes, and J. van Rheenen. Surgical implantation of an abdominal imaging window for intravital microscopy. *Nature Protocols*, 8(3):583–94, mar 2013.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, volume 9351, pages 234–241. Springer International Publishing, 2015.

H. Roth, L. Lu, A. Seff, K. Cherry, and J. Hoffman. A new 2.5 D representation for lymph node detection using random sets of deep convolutional neural network observations. In *Proc. MICCAI*, pages 520–527. Springer, Cham, 2014.

R. D. Rudyanto, S. Kerkstra, E. M. Van Rikxoort, C. Fetita, P.-Y. Brillet, C. Lefevre, W. Xue, X. Zhu, J. Liang, I. Öksüz, D. Ünay, K. Kadipas ßaoglu, R. San, J. Estépar, J. C. Ross, G. R. Washko, J.-C. Prieto, M. H. Hoyos, M. Orkisz, H. Meine, M. Hüllebrand, C. Stöcker, F. L. Mir, V. Naranjo, E. Villanueva, M. Staring, C. Xiao, B. C. Stoel, A. Fabijanska, E. Smistad, A. C. Elster, F. Lindseth, A. H. Foruzan, R. Kiros, K. Popuri, D. Cobzas, D. Jimenez-Carretero, A. Santos, M. J. Ledesma-Carbayo, M. Helmberger, M. Urschler, M. Pienn, D. G. H. Bosboom, A. Campo, M. Prokop, P. A. De Jong, C. Ortiz-De-Solorzano, A. Muñoz-Barrutia, and B. Van Ginneken. Comparing algorithms for automated vessel segmentation in computed tomography scans of the lung: the VESSEL12 study. *Medical Image Analysis*, 18:1217–1232, 2014.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training GANs. In *Proc. NIPS*, pages 2234–2242, jun 2016.

Y. Sato, S. Nakajima, and H. Atsumi. 3D Multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *In CVRMed-MRCAS'97*, 1997.

M. L. Seltzer and J. Droppo. Multi-task learning in deep neural networks for improved phoneme recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6965–6969. IEEE, may 2013.

D. Shen, G. Wu, and H.-I. Suk. Deep Learning in Medical Image Analysis. *Annual Review of Biomedical Engineering*, 19(1):221–248, jun 2017.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Proc. NIPS*, pages 802–810, 2015.

A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from Simulated and Unsupervised Images through Adversarial Training. *arXiv preprint arXiv:1612.07828*, dec 2016.

K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, pages 1–14, sep 2014.

M. Sofka and C. Stewart. Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures. *IEEE Transactions on Medical Imaging*, 25(12):1531–1546, 2006.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber. Parallel Multi-Dimensional LSTM, With Application to Fast Biomedical Volumetric Image Segmentation. *Proc. NIPS*, pages 2998–3006, 2015.

H. Su, Z. Yin, S. Huh, T. Kanade, and J. Zhu. Interactive Cell Segmentation Based on Active and Semi-Supervised Learning. *IEEE Transactions on Medical Imaging*, 35(3):762–777, mar 2016.

C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. *arXiv preprint arXiv:1707.03237*, jul 2017.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. In *Proc. NIPS*, 2014.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *Proc. CVPR*, 2015.

C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Proc. AAAI*, feb 2016.

P. Teikari, M. Santos, C. Poon, and K. Hynynen. Deep Learning Convolutional Networks for Multiphoton Microscopy Vasculature Segmentation. *arXiv preprint arXiv:1606.02382*, jun 2016.

G. M. Tozer, C. Kanthou, and B. C. Baguley. Disrupting tumour blood vessels. *Nature Reviews. Cancer*, 5(6):423–35, jun 2005.

E. Turetken, F. Benmansour, B. Andres, H. Pfister, and P. Fua. Reconstructing Loopy Curvilinear Structures Using Integer Programming. In *Proc. CVPR*, pages 1822–1829, 2013.

E. Turetken, F. Benmansour, B. Andres, P. Glowacki, H. Pfister, and P. Fua. Reconstructing Curvilinear Networks Using Path Classifiers and Integer Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(12): 2515–2530, dec 2016.

A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel Recurrent Neural Networks. *arXiv preprint arXiv:1601.06759*, jan 2016.

J. Weickert. Coherence-Enhancing Diffusion Filtering. *International Journal of Computer Vision*, 31(2-3):111–127, 1999.

R. Weinberg. *The Biology of Cancer*. Garland science, 2013.

D. H. Wolpert. The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation*, 8(7):1341–1390, oct 1996.

J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Isgum. Generative Adversarial Networks for Noise Reduction in Low-Dose CT. *IEEE Transactions on Medical Imaging*, pages 1–1, 2017.

C. Xiao, M. Staring, D. Shamonin, J. H. C. Reiber, J. Stolk, and B. C. Stoel. A strain energy filter for 3D vessel enhancement with application to pulmonary CT images. *Medical Image Analysis*, 15(1):112–124, feb 2011.

Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial Landmark Detection by Deep Multi-task Learning. In *Proc. ECCV*, 2014.

S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In *Proc. ICCV*, pages 1529–1537, 2015.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv preprint arXiv:1703.10593*, 2017.

Y. Zhu, F. Li, and T. Vadakkan. Three-dimensional vasculature reconstruction of tumour microenvironment via local clustering and classification. *Interface Focus*, 3(4):20130015, 2013.

Y. Zhu, F. Li, D. Cridebring, J. Ma, S. T. Wong, T. J. Vadakkan, M. Zhang, J. Landua, W. Wei, M. E. Dickinson, J. M. Rosen, and M. T. Lewis. Coupling Oriented Hidden Markov Random Field Model with Local Clustering for Segmenting Blood Vessels and Measuring Spatial Structures in Images of Tumor Microenvironment. In *2011 IEEE International Conference on Bioinformatics and Biomedicine*, pages 352–357. IEEE, nov 2011.