Technische Universität München
TUM School of Computation, Information and Technology

Analyzing Vascular Networks Extracted from Clinical Magnetic Resonance Angiography

Giles Tetteh

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:
Prof. Dr. Jens Großklags

Prüfer*innen der Dissertation:

1. Prof. Dr. Bjoern Menze
2. Priv.-Doz. Dr. Benedikt Wiestler

Die Dissertation wurde am 24.02.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 14.10.2023 angenommen.

ANALYZING VASCULAR NETWORKS EXTRACTED FROM CLINICAL MAGNETIC RESONANCE ANGIOGRAPHY

GILES TETTEH

Dissertation
February 2023

Giles Tetteh: *Analyzing vascular networks extracted from clinical Magnetic Resonance Angiography,* Dissertation, © February 2023

TUM School of Computation, Information and Technology
Technical University of Munich
Arcisstraße 21
80333 München
Germany

ABSTRACT

Cerebrovascular diseases are the most common life-threatening neurological events in the world. Diagnosis of this class of diseases normally involves the acquisition of angiography-based images which offer insight into the blood flow and conditions of the cerebrovascular system. Image acquisition is usually followed by processing and analysis steps which aim at extracting useful information about the structure and function of the cerebrovascular system. Common tasks include vessel segmentation and detection of the associated centerlines and bifurcation points. In the related field of natural image processing, advances in deep learning have revolutionized image analysis. With the ability to automatically learn high-level feature representation from images, deep learning methodologies have achieved outstanding performances in previously difficult tasks. The transition of deep learning methods from natural image processing to medical image processing introduces new methodological challenges due to the differences in the structure and presentation of information contained in images from these two domains.

The first part of this dissertation focuses on three main challenges that arise from using existing state-of-the-art deep learning methodologies in the analysis of vascular networks extracted from clinical data. (1) Using 3-D convolutional neural networks (CNNs) leads to a drastic increase in the number of parameters to be optimized and computations to be executed compared to 2-D CNNs. We propose cross-hair filters as a replacement for classical 3-D convolutional operations and show that the proposed cross-hair filters require less memory and are computationally faster than the classical 3-D convolutional operation. (2) Vessels account for less than 3% of the total voxels in a patient volume, centerlines represent a fraction of the segmented vessels, and visible bifurcations are in the hundredths of segmented vessels at best. This has led to the introduction of a class-balancing loss function that is computationally unstable and leads to a high false positive rate (FPR). We propose a modified loss function that is more stable and helps litigate the FPR problem. (3) There exist limited annotated data for training deep learning networks for vascular network analysis which

has stifled the pace of research in this domain. We, therefore, generate synthetic data with accurate annotations for vessel segmentation, centerlines, bifurcation points, and vessel radius. We have made the synthetic data publicly available to foster further research in this direction.

The second part of the dissertation studies collateral circulation and its grading in patients with ischaemic stroke. The quality of collateral circulation has been established as a key factor in determining the likelihood of a favorable clinical outcome and plays a major role in determining the choice of the stroke care model. Collateral flow grading by visual inspection is time-consuming and has a high inter-rater variability. We present a multi-stage deep learning approach to automate the grading process using parametric information from MR perfusion data. Experiments on different feature extraction schemes and classifiers are presented and compared to ascertain which methodology best suits the problem at hand.

ZUSAMMENFASSUNG

Zerebrovaskuläre Erkrankungen sind weltweit die häufigsten lebensbedrohlichen neurologischen Erkrankungen. Die Diagnose dieser Krankheitskategorie umfasst in der Regel die Akquisition von angiographischen Bildern, welche Aufschluss über den Blutfluss und den Zustand der Blutgefäße geben. An die Bildakquisition schließen sich in der Regel Verarbeitungs- und Analyseschritte an, die darauf abzielen, nützliche Informationen über die Struktur und Funktion des zerebrovaskulären Systems zu gewinnen. Zu den üblichen Aufgaben gehören die Segmentierung von Gefäßen und die Erkennung der zugehörigen Mittellinien und Verzweigungspunkte. Auf dem verwandten Gebiet der natürlichen Bildverarbeitung haben die Fortschritte im Bereich des Deep Learning die Bildanalyse revolutioniert. Mit der Fähigkeit, automatisch Bildmerkmale zu erlernen, haben Deep-Learning-Methoden hervorragende Leistungen bei zuvor schwierigen Aufgaben erzielt. Die Übertragung von Deep Learning Methoden der natürlichen Bildverarbeitung auf die medizinische Bildverarbeitung bringt neue methodische Herausforderungen mit sich, da sich die Struktur und die Darstellung der in Bildern aus diesen beiden Bereichen enthaltenen Informationen unterscheiden.

Der erste Teil dieser Dissertation konzentriert sich auf drei Hauptherausforderungen, die sich aus der Verwendung bestehender hochmoderner Deep-Learning-Methoden bei der Analyse zerebrovaskulärer Strukturen ergeben, die aus klinischen Daten gewonnen werden. (1) Die Verwendung von 3-D Convolutional Neural Networks (CNNs) führt zu einem drastischen Anstieg der Anzahl der zu optimierenden Parameter und der auszuführenden Berechnungen im Vergleich zu 2-D CNNs. Wir schlagen Fadenkreuzfilter als Ersatz für klassische 3-D-Faltungsoperationen vor und zeigen, dass die vorgeschlagenen Fadenkreuzfilter weniger Speicherplatz benötigen und rechnerisch schneller sind als die klassischen 3-D-Faltungsoperationen. (2) Gefäße machen weniger als 3% der gesamten Voxel eines Patientenvolumens aus, Mittellinien stellen einen Bruchteil der segmentierten Gefäße dar, und sichtbare Verzweigungen liegen bestenfalls in Hunderten von segmentierten Gefäßen vorhanden.

Dies hat zur Verwendung einer Verlustfunktion für den Klassenausgleich geführt, die rechnerisch instabil ist und zu einer hohen Falsch-Positiv-Rate (FPR) führt. Wir schlagen eine modifizierte Verlustfunktion vor, die stabiler ist und hilft, das FPR-Problem zu bewältigen. (3) Es gibt nur wenige annotierte Daten für das Training von Deep-Learning-Netzwerken für die Analyse von vaskulären Strukturen, was den Fortschritt der Forschung in diesem Bereich gebremst hat. Daher generieren wir synthetische Daten mit genauen Annotationen für die Gefäßsegmentierung, Mittellinien, Verzweigungspunkte und Gefäßradien. Wir haben die synthetischen Daten öffentlich zugänglich gemacht, um die weitere Forschung in dieser Richtung zu fördern.

Der zweite Teil der Dissertation untersucht die Kollateralen und ihre Klassifizierung bei Patienten mit ischämischem Schlaganfall. Die Qualität der Kollateralen hat sich als Schlüsselfaktor für die Wahrscheinlichkeit eines günstigen klinischen Ergebnisses erwiesen und spielt bei der Wahl des Versorgungsmodells für Schlaganfallpatienten eine große Rolle. Die Bewertung des Kollateralflusses durch visuelle Inspektion ist zeitaufwändig und weist eine hohe Interrater-Variabilität auf. Wir stellen einen mehrstufigen Deep-Learning-Ansatz vor, um die Klassifizierung unter Verwendung parametrischer Informationen aus MR-Perfusionsdaten zu automatisieren. Es werden Experimente mit verschiedenen Merkmalsextraktionsschemata und Klassifikatoren vorgestellt und verglichen, um festzustellen, welche Methodik sich am besten für das vorliegende Problem eignet.

PUBLICATIONS

This publication based thesis is based on the following peer-reviewed publications:

First author publications:

1. **Tetteh G.**, Rempfler M., Zimmer C., Menze B.H. *DeepFExt: Deep Feature Extraction for Vessel Segmentation and Centerline Prediction*. Machine Learning in Medical Imaging. MLMI 2017. Lecture Notes in Computer Science, vol 10541. Springer, Cham. 2017.

2. **Tetteh Giles**, Efremov Velizar, Forkert Nils D., Schneider Matthias, Kirschke Jan, Weber Bruno, Zimmer Claus, Piraud Marie, Menze Bjoern H. *DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes*. Frontiers in Neuroscience vol 14, 2020

3. **Tetteh Giles**, Fernando Navarro, Raphael Meier, Johannes Kaesmacher, Johannes C. Paetzold, Jan S. Kirschke, Zimmer Claus, Roland Wiest, Bjoern H Menze. *A Deep Learning Approach to Predict Collateral Flow in Stroke Patients Using Radiomic Features from Perfusion Images*. Frontiers in Neurology vol 14:1039693, 2023

The following co-author journal and conference publications are not subject to evaluation but complement the content of the above works and are reproduced in the appendix:

1. Johannes C. Paetzold, Oliver Schoppe, Rami Al-Maskari, **Giles Tetteh**, Velizar Efremov, Mihail I. Todorov, Ruiyao Cai, Hongcheng Mai, Zhouyi Rong, Ali Ertuerk, Bjoern H. Menze. *Transfer learning from synthetic data reduces need for labels to segment brain vasculature and neural pathways in 3D*. International Conference on Medical Imaging with Deep Learning, 2019.

2. Johannes C Paetzold, Suprosanna Shit, Ivan Ezhov, **Giles Tetteh**, Ali Ertürk, Bjoern Menze. *clDice-a novel connectivity-preserving loss function for vessel segmentation*. Medical Imaging Meets NeurIPS 2019 Workshop, 2019.

3. Mihail Ivilinov Todorov, Johannes Christian Paetzold, Oliver Schoppe, **Giles Tetteh**, Suprosanna Shit, Velizar Efremov, Katalin Todorov-Völgyi, Marco Düring, Martin Dichgans, Marie Piraud, Bjoern Menze & Ali Ertürk. *Machine learning analysis of whole mouse brain vasculature*. Nature Methods 17, 442-449, 2020.

In addition to above publications the following first and co-author peer-reviewed journal and conference abstracts have been published during the period of the doctoral thesis but are not included in this dissertation.

1. Esther Alberts, **Giles Tetteh**, Stefano Trebeschi, Marie Bieth, Alexander Valentinitsch, Benedikt Wiestler, Claus Zimmer, Bjoern H Menze. *Multi-modal image classification using low-dimensional texture features for genomic brain tumor recognition*. Graphs in Biomedical Image Analysis, Computational Anatomy and Imaging Genetics. GRAIL 2017, MICGen 2017, MFCA 2017. Lecture Notes in Computer Science, vol 10551. Springer, Cham.

2. Lina Xu*, **Giles Tetteh***, Mona Mustafa, Jana Lipkova, Yu Zhao, Marie Bieth, Patrick Christ, Marie Piraud, Bjoern Menze, Kuangyu Shi. *W-Net for Whole-Body Bone Lesion Detection on ^{68}Ga-Pentixafor PET/CT Imaging of Multiple Myeloma Patients*. Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment. RAMBO 2017, CMMI 2017, SWITCH 2017. Lecture Notes in Computer Science, vol 10555 23-30. Springer, Cham.

3. **G Tetteh**, A Gafita, L Xu, Y Zhao, C Dong, Axel Oliver Rominger, Kuangyu Shi, C Zimmer, BH Menze, M Eiber. *Fully Convolutional Neural Network to Assess Skeleton Tumor Burden in Prostate Cancer Using 68Ga-PSMA-11 PET/CT: Preliminary Results*. European journal of nuclear medicine and molecular imaging 45, S1, S41-S41, Springer-Verlag, 2018.

4. Cagdas Ulas, **Giles Tetteh**, Stephan Kaczmarz, Christine Preibisch, Bjoern H. Menze. *DeepASL: Kinetic Model Incorporated Loss for Denoising Arterial Spin Labeled MRI via Deep Residual Learning*. Medical Image Computing and Computer Assisted Intervention – MICCAI 2018., Lecture Notes in Computer Science, vol 11070 . Springer, Cham.

5. Cagdas Ulas, **Giles Tetteh**, Michael J Thrippleton, Paul A Armitage, Stephen D Makin, Joanna M Wardlaw, Mike E Davies, Bjoern H Menze. *Direct estimation of pharmacokinetic parameters from DCE-MRI using deep CNN with forward physical model loss*. Medical Image Computing and Computer Assisted Intervention – MICCAI 2018., Lecture Notes in Computer Science, vol 11070 39-47. Springer, Cham.

6. Yu Zhao, Hongwei Li, Rong Zhou, **Giles Tetteh**, Marc Niethammer, Bjoern H Menze. *Automatic Multi-Atlas Segmentation for Abdominal Images Using Template Construction and Robust Principal Component Analysis*. 2018 24th International Conference on Pattern Recognition (ICPR). pp 3880-3885, 2018

7. Anjany Sekuboyina, Markus Rempfler, Jan Kukačka, **Giles Tetteh**, Alexander Valentinitsch, Jan S Kirschke, Bjoern H Menze. *Btrfly Net: Vertebrae Labelling with Energy-Based Adversarial Learning of Local Spine Prior*. Medical Image Computing and Computer Assisted Intervention – MICCAI 2018. MICCAI 2018. Lecture Notes in Computer Science, vol 11073. Springer, Cham.

8. Yu Zhao, Hongwei Li, Shaohua Wan, Anjany Sekuboyina, Xiaobin Hu, **Giles Tetteh**, Marie Piraud, Bjoern Menze *Knowledge-Aided Convolutional Neural Network for Small Organ Segmentation.* IEEE Journal of Biomedical and Health Informatics, vol. 23, no. 4, pp. 1363-1373, July 2019.

9. Andrei Gafita, Marie Bieth, Markus Krönke, **Giles Tetteh**, Fernando Navarro, Hui Wang, Elisabeth Günther, Bjoern Menze, Wolfgang A Weber, Matthias Eiber. *qPSMA: semiautomatic software for whole-body tumor burden assessment in prostate cancer using ^{68}Ga-PSMA11 PET/CT*. Journal of Nuclear Medicine 2019, 60 (9) 1277-1283.

10. Yu Zhao, Andrei Gafita, Bernd Vollnberg, **Giles Tetteh**, Fabian Haupt, Ali Afshar-Oromieh, Bjoern Menze, Matthias Eiber, Axel Rominger, Kuangyu Shi. *Deep neural network for automatic characterization of lesions on 68 Ga-PSMA-11 PET/CT*. European Journal of Nuclear Medicine and Molecular Imaging volume 47, pp 603–613 (2020).

11. Florian Kofler, Christoph Berger, Diana Waldmannstetter, Jana Lipkova, Ivan Ezhov, **Giles Tetteh**, Jan Kirschke,

Claus Zimmer, Benedikt Wiestler, Bjoern H Menze. *BraTS Toolkit: translating BraTS brain tumor segmentation algorithms into clinical and scientific practice*. Frontiers in neuroscience, vol 14 pp 125 2020.

12. Stefan Gerl, Johannes C Paetzold, Hailong He, Ivan Ezhov, Suprosanna Shit, Florian Kofler, Amirhossein Bayat, **Giles Tetteh**, Vasilis Ntziachristos, Bjoern Menze. *A distance-based loss for smooth and continuous skin layer segmentation in optoacoustic images*. Medical Image Computing and Computer Assisted Intervention – MICCAI 2020. MICCAI 2020. Lecture Notes in Computer Science, vol 12266 pp 309-319. Springer, Cham.

13. Anjany Sekuboyina, Malek E Husseini, Amirhossein Bayat, Maximilian Löffler, Hans Liebl, Hongwei Li, **Giles Tetteh**, et al. *VerSe: a vertebrae labelling and segmentation benchmark for multi-detector CT images*. Medical image analysis 73 pp 102166 Elsevier, 2021.

14. Maximilian T Löffler, Alina Jacob, Andreas Scharr, Nico Sollmann, Egon Burian, Malek El Husseini, Anjany Sekuboyina, **Giles Tetteh**, Claus Zimmer, Jens Gempt, Thomas Baum, Jan S Kirschke. *Automatic opportunistic osteoporosis screening in routine CT: improved prediction of patients with prevalent vertebral fractures compared to DXA*. Eur Radiol 31, 6069–6077 (2021).

* Joint first authorship

For a long time it had seemed to me that life was about to begin - real life. But there was always some obstacle in the way, something to be gotten through first, some unfinished business, time still to be served, a debt to be paid. Then life would begin. At last it dawned on me that these obstacles were my life.

— Alfred D. Souza

ACKNOWLEDGMENTS

"We are like dwarfs on the shoulders of giants, so that we can see more than they, and things at a greater distance, not by virtue of any sharpness of sight on our part, or any physical distinction, but because we are carried high and raised up by their giant size" – Bernard De Chartres.

My sincere gratitude goes to my supervisors Prof. Dr. Bjoern H. Menze and Prof. Claus Zimmer for their patience, guidance, the freedom to be innovative, and for the opportunity to work on such an interesting topic. A big thank you to all my clinical collaborators from Klinikum Recht der Isar Munich and Inselspital Universitätsspital Bern especially Prof. Dr. Jan S. Kirschke, Prof. Dr. Kuangyu Shi, and Dr. Raphael Meier for the amazing clinical inputs and discussions during the later part of my works.

I would also like to thank PD. Dr. Benedikt Wiestler for his kind offer to review this work as an examiner as well as Prof. Jens Großklags for chairing the examining committee.

To all my office and lab colleagues over the years at IBBM and MRI Neuroradiology, thank you for the fruitful discussions, collaborations, and the many lunch breaks we had together – in particular, Alexander Valentinitsch, Amir Hossein Bayat, Anjany Sekuboyina, Augusto F. Sanches, Cagdas Ulas, Carolin Pirkl, Dhritiman Das, Diana Waldmannstetter, Esther J. Alberts, Fernando Navarro, Florian Kofler, Hongwei (Bran) Li, Ivan Ezhov, Jana Lipkova, Johannes Paetzold, John LaMaster, Judith Zimmermann, Lina Xu, Malek El Husseini, Marie Bieth, Marie Piraud, Markus Rempfler, Oliver Schoppe, Rami Al-Maskari, Sebastian Rühling, Suprosanna Shit, Yu Zhao.

Thank you to the past and immediate staff of TUM-GCB and

TUM-GS for your support with all administrative matters and for coordinating many scientific and transferable skills seminars. Particularly Julia Knürr (Msc.) and Dr. Anja Drescher for welcoming me anytime I approach them with administrative issues.

Several sources of funding made this work possible (see funding sources listed in publications), but in particular, I want to acknowledge the direct grant support that I received over the first years of this work from the Deutscher Akademischer Austauschdienst (DAAD).

Most importantly, to my family and friends in Ghana, Germany, and beyond – Thank you for making this journey worthwhile. Especially my wife Jacqueline, my parents Veronica and Polycarp, and my siblings Theresa, Clara, Joana, Callista, and Jeshiba for all the love and support. Thank you to Gideon Clottey and Edward Aggrey of Impact Global Consult for being a brother and supporting this adventure. To my Godmother Dolores Opon (Ph.D.), thank you for believing in me. Thank you also to Prof. (Bro.) Michael Amakyi for being a good mentor and lending a helping hand any time it is needed. Special thank you to the Ghana community in Munich (GHStuff) for the amazing socialization events which made me feel at home.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CIFAR	Canadian Institute For Advanced Research
CNN	Convolutional Neural Network

COCO	Common Objects in Context
CT	Computed Tomography
CTA	Computed Tomography Angiogrphy
DRIVE	Digital Retinal Images for Vessel
DSA	Digitally Subtracted Angiography
FCNN	Fully Convolutional Neural Network
FPR	False Positive Rate
GAN	Generative Adversarial Network
HU	Hounsfield Unit
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
MNIST	Modified National Institute of Standard and Technology
MR	Magnetic Resonance
MRA	Magnetic Resonance Angiography
PASCAL	Pattern Analysis, Statistical Modelling and Computational Learning
rBF	relative Blood Flow
rBV	relative Blood Volume
ROI	Region of Interest
STARE	STructured Analysis of the REtina
VAE	Variational Auto-encoder
VesSAP	Vessel Segmentation and Analysis Pipeline

Part I

INTRODUCTION AND BACKGROUND

1 INTRODUCTION

1.1 MOTIVATION AND OPEN CHALLENGES

Angiography offers insights into the blood flow and conditions of the vascular tree. Three dimensional volumetric angiography information can be obtained using magnetic resonance (MRA), ultrasound, or x-ray based technologies like computed tomography (CT). A common first step in analyzing these data is vessel segmentation. Still, moving from raw angiography images to vessel segmentation alone might not provide enough information for clinical use, and other vessel features like centerline, diameter, or bifurcations of the vessels are also needed to accurately extract information about the vascular tree, for example, to characterize its structural properties or flow pattern.

The first sets of public data for vessel segmentation were the STructured Analysis of the Retina (STARE) [19] and Digital Retinal Images for Vessel (DRIVE) [47] datasets. These datasets were 2-D in nature and served as the gold standard for testing computer vision algorithms on vessel segmentation. Early works on these datasets made use of nearest neighbor, kernel boost, random forest, and other classical machine learning algorithms [3, 13, 34] and achieved results comparable to a 2nd human annotator. Motivated by the success of deep learning in natural images, the focus was shifted to deep learning algorithms which further improved results over human annotators. Maninis et al. [30] proposed a unified framework of CNN for retinal image analysis that provides both retinal vessel and optic disc segmentation and achieved a dice score of 82% compared to a human performance of 79% on the DRIVE dataset and 84% dice score compared to human annotator score of 76% on the STARE dataset.

Though vessel segmentation in the STARE and Drive datasets is successful, medical images obtained from modalities like MRA and CT are 3-D in nature. Adopting the existing methodologies to handle the extra dimension introduces technical and implementation problems. In addition, vessel segmentation, cen-

terline prediction, and bifurcation detection tasks which are the focus of this work introduce even more specialized challenges enumerated below:

- **Memory and computational limitation**: Processing of 3-D medical volumes poses a memory consumption and speed challenge. Using 3-D convolutional neural networks (CNNs) leads to a drastic increase in the number of parameters to be optimized and computations to be executed when compared to 2-D CNNs. This poses restrictions on the depth and kernel sizes that can be employed on medical data given the currently limited memory and computational resources. At the same time, applying a 2-D CNN in a slice-wise fashion discard valuable 3-D context information that is crucial for tracking curvilinear structures in 3-D.

- **Extreme class imbalance**: In most medical image analysis tasks the object of interest is less than 5% of the overall data points leading to a general problem of class imbalance. The vessel, centerline, and bifurcation prediction tasks are especially characterized by high class imbalances. Vessels account for less than 3% of the total voxels in a patient volume, centerlines represent a fraction of the segmented vessels, and visible bifurcations are in the hundredths at best – even when dealing with volumes with 10^6 and more voxels. This bias toward the background class is a common problem in medical data [5, 16, 17]. To achieve a balanced prediction power between the foreground and background one has to apply a class balancing loss function during the training of deep learning networks. Unfortunately, current class balancing loss functions for training CNNs turn out to be numerically unstable in extreme cases such as vessel segmentation, centerline prediction, and bifurcation detection.

- **Lack of annotated data for training**: Deep learning has been successful with natural images partly due to the existence of large-scale public datasets for training and evaluation. Publicly available data for medical image analysis is however very limited partly due to special regulations concerning medical data and the fact that manually annotating medical data, in general, is tedious and requires professional expertise. Vessel, centerline, and bifurcation annotation in particular require even more hours of work

and expertise. This has stifled the pace of research in medical image analysis in general and especially brain vessel analysis.

1.2 CONTRIBUTIONS

This dissertation is made up of two main parts. The first part, captured in two peer-reviewed publications [49, 51], contributes to the advancement of research in the analysis of vascular networks extracted from clinical MRA data using deep learning. We identify challenges associated with applying deep learning methods to the tasks of vessel segmentation, centerline prediction, and bifurcation detection in medical data. We then propose, implement and test solutions to the identified challenges. The technical and scientific contributions comprise the following:

- proposing a lightweight deep feature extraction (deepFExt) scheme for extracting multi-level and multi-scale image features crucial for detecting and tracking curvilinear structures in medical images.
- developing fast cross-hair filters, a substitute for the classical 3-D convolutional kernels which offer improved speed and memory usage while maintaining accuracy.
- proposing class balancing scheme and false prediction (FP) rate correction term for training deep neural networks in tasks where there is an extreme class imbalance between the object of interest and the background.
- generating synthetic data, through a simulation of a vascular tree that follows a generative process inspired by the biology of angiogenesis, with annotations for vessel, centerlines, and bifurcation points and making it possible to train deep networks for these tasks with few expert annotations.

The second part of this dissertation, captured in the journal publication Tetteh et al. [50], studies collateral circulation and its grading in ischaemic stroke patients. The quality of collateral circulation has been established as a key factor in determining the likelihood of a favorable clinical outcome and goes a long way to determining the choice of the stroke care model. The current method of grading collateral flow in clinical routine is through

visual inspection. This is time consuming and introduces bias in the assigned grade. This dissertation presents a multi-stage deep learning approach to automate the grading process. Experiments on different feature extraction schemes and classifiers are presented and compared to ascertain which methodology best suits the problem at hand. This serves as a first attempt at fully automating the collateral flow grading process and will serve as a basis for further research in this direction.

1.3 OUTLINE

In part I, we have already been introduced to the motivation of this dissertation and we have also seen an overview of the two main contributions of this work. The following chapter 2 provides background to medical image processing, highlights the research journey and contributions at each stage of the journey, and is divided into six sections. Sec. 2.1 gives a brief history of medical image processing and the migration from natural image processing to medical image analysis in the deep learning domain. Sections 2.2, 2.3, and 2.4 cover some of the challenges associated with using deep learning in medical image analysis and proposed methodological solutions in this work. Sec. A.3 discusses a follow-up work on the analysis of whole mouse brain vasculature which uses the methodological improvements proposed in this dissertation. Sec. 2.6 gives a brief introduction to the work on collateral flow grading some of the main findings.

In part II, the main publications considered under this publication-based dissertation are presented. Chapter 3 [51] presents our work on extracting deep features using inception models. We analyze the extracted features and use these features as input to a deep learning pipeline to segment vessels and predict centerlines in retinal images. In Chapter 4 [49] we present thoroughly the proposed methodological improvements aimed at handling the challenges associated with applying deep learning to brain vascular analysis and medical image processing in general. We formulate the cross-hair filter as an efficient replacement for the classical 3-D convolutional operation, propose a stable loss function, and generate synthetic data which can be used for transfer learning purposes in the brain vessel analysis domain. In Chapter 5 [50], we discuss challenges associated with collateral flow grading and present a set of tools aimed at automating the grading process. We experiment with different methodolo-

gies in the form of feature extractors and classifiers and compare the outcome to know which methodology suit the problem.

In part III, the overall contribution of this dissertation and the important findings are discussed in Chapter 6. We follow up with the conclusion and outlook for future work in Chapter 7 where we discuss what will be interesting research that can spawn out of this dissertation.

The appendix in part IV includes relevant co-author publications which are not considered in the evaluation of this publication-based dissertation but complement the contributions of this dissertation in a form of follow-up studies or tests of the proposed methodological improvements.

2

BACKGROUND

2.1 MEDICAL IMAGE PROCESSING

Medical image processing and analysis cover a range of potential topic areas including image acquisition, image enhancement, image compression and storage, image analysis, and image visualization. The area of image analysis in the medical domain includes the development of methodologies for tasks like image segmentation, image registration, motion tracking and change detection in image sequences, and the measurement of anatomical and physiological parameters from images. These research efforts are direct responses to system-oriented problems like image-guided surgery/intervention, atlas-based description of entire anatomical regions, deformation analysis based on biomechanical and other models, and visualization of anatomical and physiological processes [10]. Research back in the 1970s treated medical and biomedical image analysis as a unique information processing problem, one where approaches based on pattern recognition, image/signal processing, and computer vision may play a role [10]. Ballard and Sklansky [1, 2] used computer vision-based methods to enhance the edges of the images of tumors in radiographs and isotope scans to ease the tumor detection task and a ladder-structured decision tree for recognizing tumors in chest radiographs. Pizer and Todd-Pokropek [35] Also used computer processing to improve the quality of scintigrams and increase the accuracy with which the image approximates the activity distribution by reversing degradation. In the 1990s, research in computer vision took a new turn with machine learning and deep learning at the center stage. Most researchers in the field at the time focused on the processing and analysis of natural images. After about a decade, informed by the success stories in natural image analysis, early attempts at applying deep learning in medical image analysis started.

2.1.1 *Natural Images to Medical Image Analysis*

Computer vision is an interdisciplinary field that enables systems and computers to derive resourceful information from

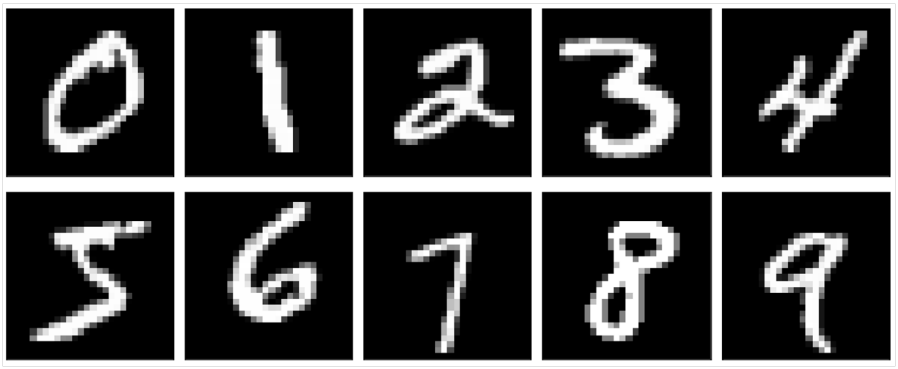


Figure 2.1: Sample of images from the MNIST dataset. There are a total of 60,000 examples in the training set and 10,000 examples in the test set. Each image has a resolution of 28×28 pixels.

videos, digital images, and other forms of visual inputs. In the early stages of computer vision and image processing, the interest was on object recognition or *what can be seen* in a given image and requires primarily assigning a single class label to a given image or video frame.

Very common among these tasks is the classification of handwritten digits from the Modified National Institute of Standard and Technology (MNIST) [25] (Sample images in Figure 2.1). From an initial error rate of 12% [25] in the first attempts in the late 1990s to an error rate of less than 0.5% [7, 8, 31, 45] in a span of 10 years the computer vision community was ready to take up more challenging tasks.

Focus was quickly shifted towards the classification of natural images with two databases from the Canadian Institute For Advanced Research (CIFAR) - CIFAR-10 and CIFAR-100 [23] which are labeled subsets of 80 million tiny $(32px \times 32px)$ images with 10 and 100 classes respectively being one of the first attempts at the task. Sample images from the CIFAR-10 dataset are presented in Figure 2.2. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9] followed with a dataset of over 10 million images in 1000 categories (Sample images in Figure 2.3).

With the success in natural image classification, there is high interest in going beyond just what can be seen in an image to answer the question of "where can it be seen?". This led to object localization, detection, and semantic segmentation tasks. Figure

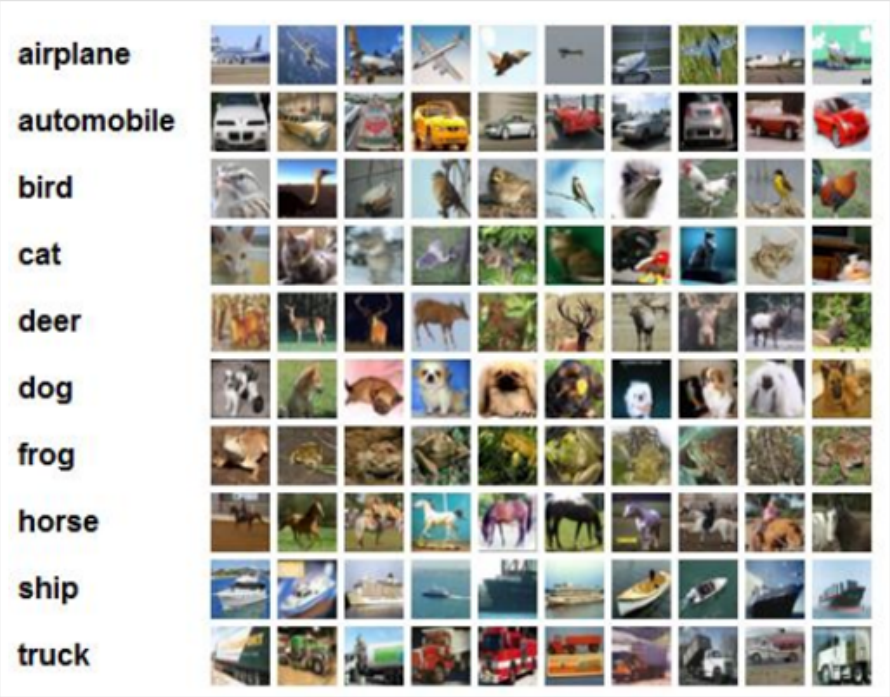


Figure 2.2: Sample of images from the CIFAR-10 dataset. There are 10 classes with 6,000 images per class. A total of 50,000 examples in the training set and 10,000 examples in the test set. Each image has a resolution of 32×32 pixels.

2.4 shows an overview of the category of computer vision tasks that goes beyond "what can be seen" to identify where the object can be seen within the image. Very common in this category of computer vision tasks are the Caltech [15], PASCAL [11], LabelMe [39], TinyImages [53], ImageNet [9], SUN [54], and COCO [28] datasets with single and multiple instance object detection and localization. These and other large-scale datasets paved the way for rapid growth in the field of categorical object detection by providing rich training and evaluation datasets and leading to the development of more robust learning algorithms and architectures.

Different versions of CNN-based architectures have been proposed for handling object detection tasks. Krizhevsky, Sutskever, and Hinton [24] proposed AlexNet, a deep convolutional neural network, to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes and achieved top-1 and top-5 error rates of 37.5% and 17.0% respectively. Simonyan and Zisserman [46] investigated the effect of the convolutional network depth on its accuracy in the

Figure 2.3: Samples of images from the ImageNet dataset. The dataset contains over 14 million annotated images for image classification and object detection.

large-scale image recognition setting and proposed the VGGNet architecture, which is made up of 16-19 convolutional layers and overperforming the state-of-the-art at the time. An inception model based on the idea of a network within a network was examined by Szegedy et al. [48] leading to 22 layers deep network called GoogLeNet. He et al. [18] explicitly reformulated network layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions and proposed a Residual Network (ResNet) architecture with a depth of up to 152 layers which achieved 3.57% error rate on the ImageNet dataset - an over 40% improvement in the state-of-the-art at the time. The trend in these groundbreaking architectures has been robust data representation and network depth.

Based on the above success stories, early attempts at applying deep learning to medical image analysis started. Prasoon et

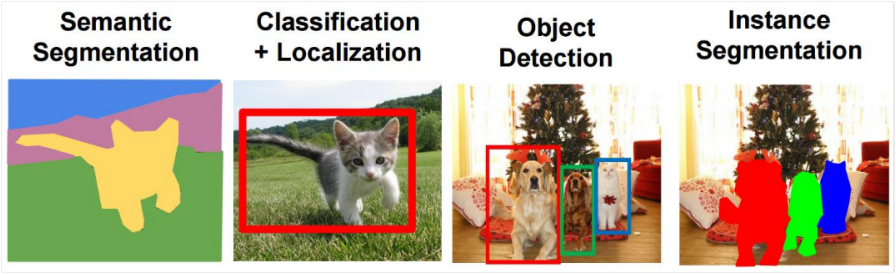


Figure 2.4: Computer vision tasks that go beyond "what can be seen" to the detection of objects of interest through object detection and classification, semantic segmentation, and instance segmentation (Li, Johnson, and Yeung [26]).

al. [36] and Liao et al. [27] used CNNs for segmentation of knee cartilage and prostate in MR scans respectively. Ronneberger, Fischer, and Brox [37] proposed U-NET, a network architecture consisting of a contracting path to capture context and a symmetric expanding path that enables precise localization. The U-NET architecture was used for the segmentation of neuronal structures in electron microscopic stacks and also won the ISBI cell tracking challenge 2015 by a large margin. U-NET and 3D U-NET [6] inspired a lot of pixel and voxel-based architectures and were modified and applied to several medical image segmentation tasks [5, 20, 32, 56–59] with Siddique et al. [44] providing a comprehensive review on U-NET and its variants for medical image segmentation.

Extending deep learning methods from natural images to medical images, in general, requires one to consider the main differences between natural images and medical images.

- Medical images have increased dimension and resolution; images are mostly 3-D in nature which implies that computations and memory requirements are increased exponentially. Again, medical images are in most cases of higher resolution compared to real-world images which again adds to the memory and computational problem of the third dimension. Medical images capture a wider range of the light spectrum as compared to natural images and hence have a higher intensity range. Most natural images have and intensity of integer data type ranging between 0-255 per pixel which requires an 8bits (2^8) memory per pixel for storage. Medical images on the other hand have completely different intensity ranges and data types. Im-

ages from CT scanners can have intensities between -1024 to $+3071$ HU, and sometimes single or double precision floating point data types.

- Another profound challenge with medical image analysis is the limited data for training due to regulatory requirements. The limited available data are also not open sourced and therefore benchmark testing and thorough algorithm performance comparison are not possible. Unlike natural images, acquiring annotations for medical data requires professional expertise and time. Some tasks require annotation to be carried out in 3-D and on a voxel level which is either impossible or takes longer professional hours from medical practitioners.

- In addition to the above differences, vascular analysis in medical images which is the focus of this works places more relevance on local information than global information. Vessel segmentation, centerline prediction, and bifurcation detection tasks require voxel-level details, and therefore downsampling the images as a way of conserving memory can have a detrimental effect on the final accuracy of these tasks due to the loss in the low-level information.

These differences lead to different challenges and require medical images to be handled differently to achieve the best results from existing methodologies and computational resources. In Sections 2.2,2.3, and 2.4, we discuss some of the challenges arising from these differences and propose solutions in the form of a memory-efficient cross-hair filters, a class balancing loss function, and a synthetically simulated data for transfer learning.

2.2 MEMORY EFFICIENT CROSS-HAIR FILTERS

We will start our discussion on the proposed cross-hair filters by formulating the memory and computational requirements for standard 2-D and 3-D convolutional operations. Given an image I, and a kernel M of size a convolutional operator $*$ can be defined for 2-D and 3-D cases respectively as follows:

$$I * M = A = \{a_{ij}\}; a_{ij} = \sum_{r=1}^{k_x} \sum_{s=1}^{k_y} I_{(R,S)} M_{(r,s)} \tag{2.1}$$

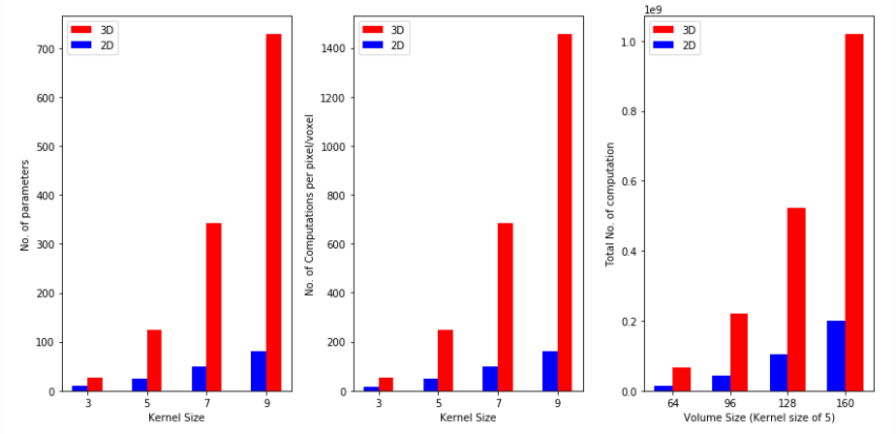


Figure 2.5: A graph showing the computational and memory consumption of 2-D vs 3-D convolutional operations. Blue bars represent 2-D operations and red bars represent 3-D operations. The left graph shows memory consumption in the form of the number of parameters needed to store in memory during the computation for a single convolutional operation given the kernel size. The middle graph represents the number of computations per voxel/pixel given the kernel size. The right graph shows the total number of computations in billions ($1e^9$) for a given image size when we use a kernel size of five (5).

$$I * M = A = \{a_{ijk}\}; a_{ijk} = \sum_{r=1}^{k_x} \sum_{s=1}^{k_y} \sum_{t=1}^{k_z} I_{(R,S,T)} M_{(r,s,t)} \tag{2.2}$$

where $R = i + r - (1 + [\frac{k_x}{2}])$, $S = j + s - (1 + [\frac{k_y}{2}])$, and $T = k + t - (1 + [\frac{k_z}{2}])$.

The above equation involves $k_x k_y k_z$ multiplications and $k_x k_y k_z - 1$ additions for each voxel of the resulting image A in the 3-D case as compared to $k_x k_y$ multiplications and $k_x k_y - 1$ additions in the 2-D case. Figure 2.5 shows the computational and memory requirements in different scenarios for 2-D and 3-D convolutional operations and depicts clearly the exponential increase in computational and memory needs between 2-D and 3-D operations especially when we have increased image and kernel sizes which is normally the case for medical image analysis.

The immediate solutions to the computational and memory problem have been applying 2-D convolutions to the slices of

the 3-D volumes and stitching the results of the 2-D slices in a post-processing step to get back the 3-D volume [5, 29, 42]. Other works improve this by feeding neighboring voxels as additional channels for the 2-D convolution [38]. The main setback to these approaches is the lack of 3-D context information which is crucial when dealing with vessel segmentation, centerline prediction, and bifurcation detection tasks due to the size of the objects of interest.

2.2.1 *Cross-hair filter for 2-D convolutional operations*

In other to retain context information and at the same time address the computational and memory problem we propose the use of cross-hair filters which involve convolution along the perpendicular axis of the center point of interest. To demonstrate this we start with 2-D cross-hair filters where we replace normal 2-D filters with one that uses information only along the perpendicular axis. Figure 2.6 is a graphical depiction of the proposed cross-hair filter for 2-D image processing. By introducing the cross-hair filters we reduce the parameter size from $k_x k_y$ to $(k_x + k_y)$ and the number of computations from $2k_x k_y - 1$ to $2(k_x + k_y) - 1$ and Figure 2.7 shows these gains in computation and memory.

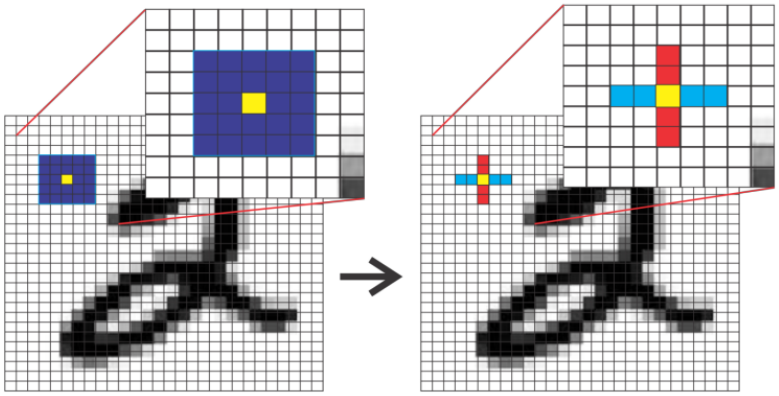


Figure 2.6: Proposed 2-D cross-hair filters to replace full 2-D kernels. At the left is a depiction of the full 2-D kernels in blue and at the right is the cross-hair filters with light-blue and red representing the perpendicular axis of interest.

To test the proposed cross-hair filters we perform a preliminary experiment using the MNIST handwritten digit data. We designed a CNN with three convolutional layers of kernel size

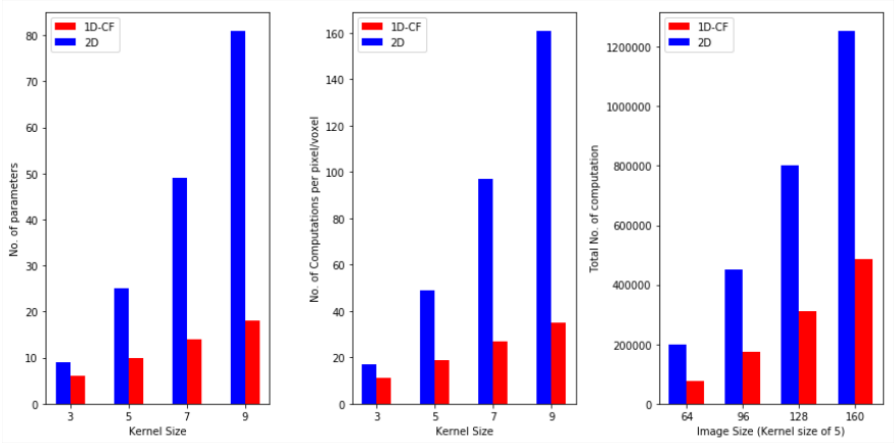


Figure 2.7: A graph showing the computational and memory consumption of full 2-D vs cross-hair filters(1D-CF). Blue bars represent 2-D operations and red bars represent cross-hair operations. The left graph shows memory consumption in the form of the number of parameters needed to store in memory during the computation for a single convolutional operation given the kernel size. The middle graph represents the number of computations per pixel given the kernel size. The right graph shows the total number of computations for a given image size when we use a kernel size of five (5).

(5×5) and three hidden layers. Details of the experiment are provided in Figure 2.8. By replacing the full 2-D kernels with cross-hair filters in the convolutional layers we can improve the computational speed by about 18% (from Table 2.1) while at the same time achieving comparable accuracy.

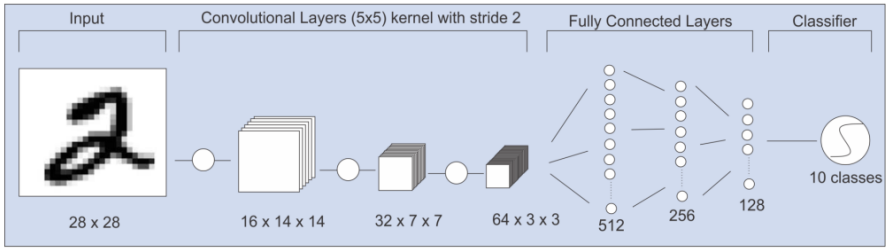


Figure 2.8: Experimental setup of the preliminary test of the 2-D cross-hair filter. To make comparison we run two experiments with the same network structure but replace the 2-D convolutional kernels with the proposed cross-hair kernels in one of the experiments.

Table 2.1: Quantitative results from preliminary experiments on the MNIST handwritten digit classification using full 2-D and cross-hair filters. The result shows comparable results in terms of error and an improvement of about 18% in execution time.

Method	Error	Execution Time
Full 2-D	1.27%	$4.67e^{06}$
Cross-hair	1.58%	$3.79e^{06}$

2.2.2 *Cross-hair filter for 3-D convolutional operations*

With the success of the preliminary experiments, we proceed to formulate the cross-hair filter for 3-D convolutional operations. By replacing the full 3-D convolution with convolutions along the three perpendicular planes we can reformulate the Equation 2.2 as follows:

$$a_{ijk} = \alpha \sum_{s=1}^{k_y} \sum_{t=1}^{k_z} I_{(i,S,T)} M^{i}_{(s,t)} + \beta \sum_{r=1}^{k_x} \sum_{t=1}^{k_z} I_{(R,j,T)} M^{j}_{(r,t)} + \gamma \sum_{r=1}^{k_y} \sum_{s=1}^{k_y} I_{(R,S,k)} M^{k}_{(r,s)} \tag{2.3}$$

where M^i, M^j, and M^k are 2-D filters extracted from the original 3-D filter M and α, β, and γ are plane weights. This reduces the number of parameters from $k_x k_y k_z$ to $(k_x k_y + k_x k_z + k_y k_z)$ and the number of elementary operations (computations) from $2(k_x k_y k_z) - 1$ to $2(k_x k_y + k_x k_z + k_y k_z) - 1$. We can easily see that $(k_x k_y + k_x k_z + k_y k_z) \leq (k_x k_y k_z)$ for $k_x, k_y, k_z \geq 3$ which is normally the case for convolutional operations in image processing.

Figure 2.9 shows improved computation and memory requirements between full 3-D filters and the proposed cross-hair filters. We conduct further experiments to test the actual improvements in speed and memory consumption with the VNET architecture proposed by Milletari, Navab, and Ahmadi [32]. With a volume size of $(128 \times 128 \times 128)$ and by replacing the full 3-D filters with our proposed 2-D filters we reduce the number of parameters from 22.89×10^6 to 16.56×10^6 and the execution time from 11 to 7 seconds which represent at least 27% improvement in both measures. Details of the experimental setup and full results of this experiment are presented in Tetteh et al. [49]. The results from the experiments show that cross-hair filters are an efficient

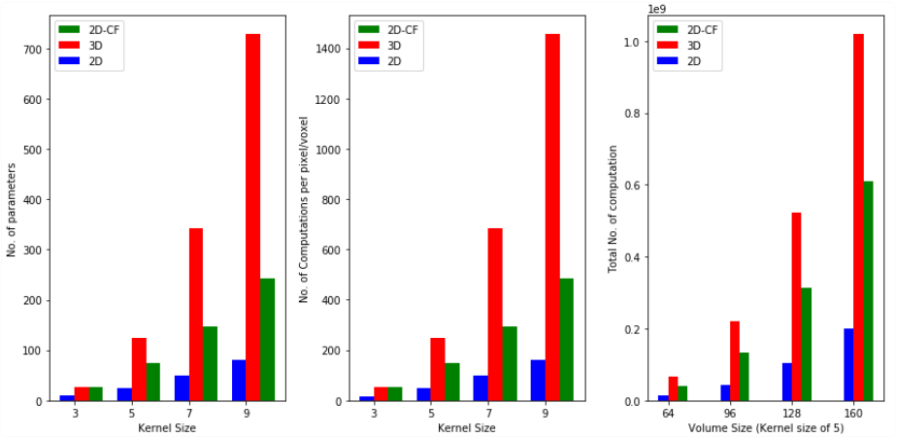


Figure 2.9: Comparison of computational and memory requirements of full 2-D, full 3-D, and cross-hair filters. The left graph shows the number of parameters given per kernel size. The middle graph shows the number of computations per voxel for the given kernel size and the right graph shows the total number of computations in billions ($1e^9$) for the given volume size and a kernel size of five (5).

way of performing 3-D convolutional operations while at the same time maintaining enough 3-D context information which improves accuracy over 2-D convolutional operations.

2.3 EXTREME CLASS BALANCING LOSS FUNCTION

In most medical image processing tasks, the object of interest represents less than 5% of the overall volume. Especially in segmentation tasks, there is an extreme class imbalance between the background and the object of interest. For our specific tasks of vessel segmentation, centerline prediction, and bifurcation detection the problem increases with a maximum of less than 3% of vessels compared to the background. Figure 2.10 shows the label distribution in the datasets used in experiments presented in Tetteh et al. [49]. This is an issue when training a deep network because by predicting the whole image as background the network is already achieving over 97% accuracy. This makes the network learn to predict the background at the expense of the objects of interest when trained with the standard cross-entropy loss proposed in the literature below.

$$\mathcal{L}(\mathbf{W}) = -\frac{1}{N}[\sum_{j \in Y_+} \log P(y_j = 1|X;\mathbf{W}) + \sum_{j \in Y_-} \log P(y_j = 0|X;\mathbf{W})]$$

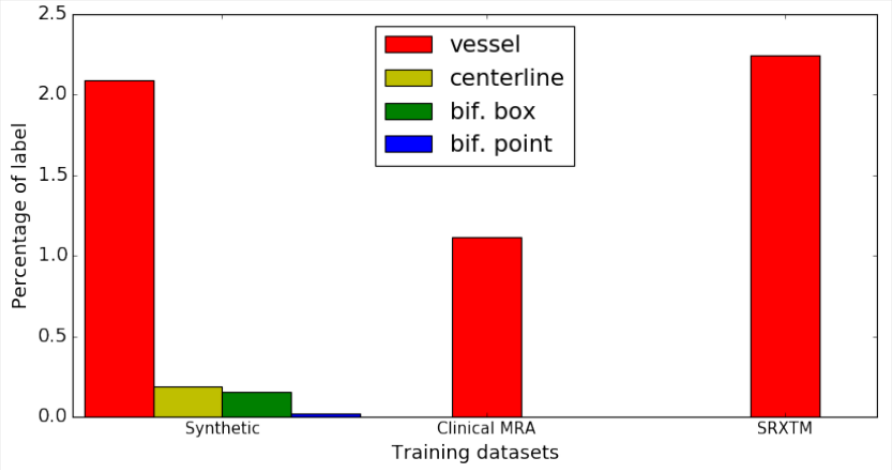


Figure 2.10: Distribution of foreground labels in the three data sets used in Tetteh et al. [49]. Red represents the percentage of vessel labels, yellow represents the percentage of centerline labels, blue represents bifurcation points and green represents bifurcation points with a $(3 \times 3 \times 3)$ cube around it

(2.4)

where N is the total number of labels, Y_+ is the set of foreground labels, Y_- is the set of background labels, $P(.)$ is the probability operator, X is the given image, and $\mathbf{W}$ is the set of network parameters.

The solution is the introduction of class weights into the loss functions used during the training phase to penalize the network more when it predicts the foreground as the background. Earlier works by [5, 30, 55] proposed the use of a parameter β which is the ratio of the number of background labels to the total labels to penalize the network for wrong foreground prediction and $1 - \beta$ for wrong background prediction. This transforms the standard cross-entropy loss function in Equation 2.4 to the form below.

$$\mathcal{L}(\mathbf{W}) = -\beta \sum_{j \in Y_+} \log P(y_j = 1|X;\mathbf{W}) - (1-\beta) \sum_{j \in Y_-} \log P(y_j = 0|X;\mathbf{W}) \tag{2.5}$$

where $\beta = \frac{|Y_-|}{|Y|}$, $|Y_-|$ is the number of background labels and $|Y|$ is the total number of labels.

Our experiments with the class-balancing weights in Equation 2.5 however show that the weights β and $1-\beta$ as defined above first of all lead to numerical instability for big datasets and also introduce a problem of high false positive rate. We, therefore, propose new weights to correct the numerical instability and a new term ($\mathcal{L}_2(\mathbf{W})$) to address the problem of high false positive rate.

$$
\begin{aligned}
\mathcal{L}(\mathbf{W}) &= \mathcal{L}_1(\mathbf{W}) + \mathcal{L}_2(\mathbf{W}) \qquad (2.6)\\
\mathcal{L}_1(\mathbf{W}) &= -\frac{1}{|Y_+|}\sum_{j\in Y_+}\log P(y_j=1|X;\mathbf{W})\\
&- \frac{1}{|Y_-|}\sum_{j\in Y_-}\log P(y_j=0|X;\mathbf{W})\\
\mathcal{L}_2(\mathbf{W}) &= -\frac{\gamma_1}{|Y_+|}\sum_{j\in Y_{f+}}\log P(y_j=0|X;\mathbf{W})\\
&- \frac{\gamma_2}{|Y_-|}\sum_{j\in Y_{f-}}\log P(y_j=1|X;\mathbf{W})\\
\gamma_1 &= 0.5+\frac{1}{|Y_{f+}|}\sum_{j\in Y_{f+}}|P(y_j=0|X;\mathbf{W})-0.5|\\
\gamma_2 &= 0.5+\frac{1}{|Y_{f-}|}\sum_{j\in Y_{f-}}|P(y_j=1|X;\mathbf{W})-0.5|
\end{aligned}
$$

where Y_{f+} and Y_{f-} are the set of false positive and false negative predictions respectively.

Here $\mathcal{L}_1(\mathbf{W})$ is a more numerically stable version of Equation 2.5 and $\mathcal{L}_2(\mathbf{W})$ is design to control the false positive rate. We perform further experiments to compare the proposed loss function in Equation 2.6 with the existing loss function in Equation 2.5 and our results as depicted in Figure 2.11 shows that training with $\mathcal{L}_2$ improves the ratio of precision to recall (a value closer to 1) compared to training without $\mathcal{L}_2$.

2.4 SYNTHETIC DATA FOR TRANSFER LEARNING

One of the major reasons for the success of natural image processing is the availability of large-scale datasets. Deeper networks require more data for training and testing which is a major challenge with medical data. Medical data are hardly open-sourced due to stringent regulations. In limited cases where

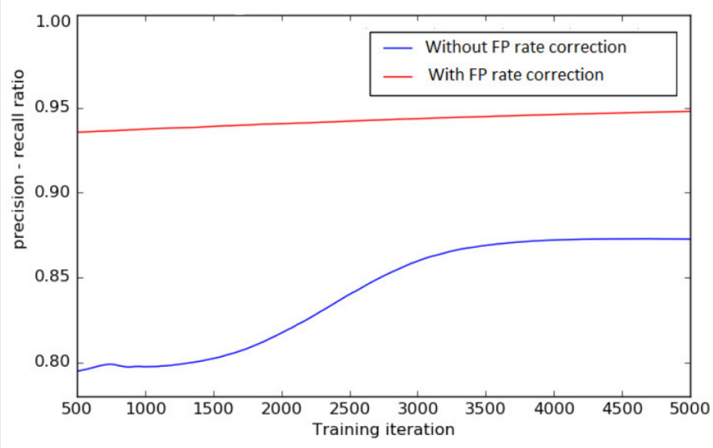


Figure 2.11: Precision-recall ratio during training, with false positive (FP) rate correction $\mathcal{L}_2$ and without FP rate correction. The graph shows that FP rate correction (red line) leads to a higher and more stable ratio compared with training without FP rate correction (blue line).

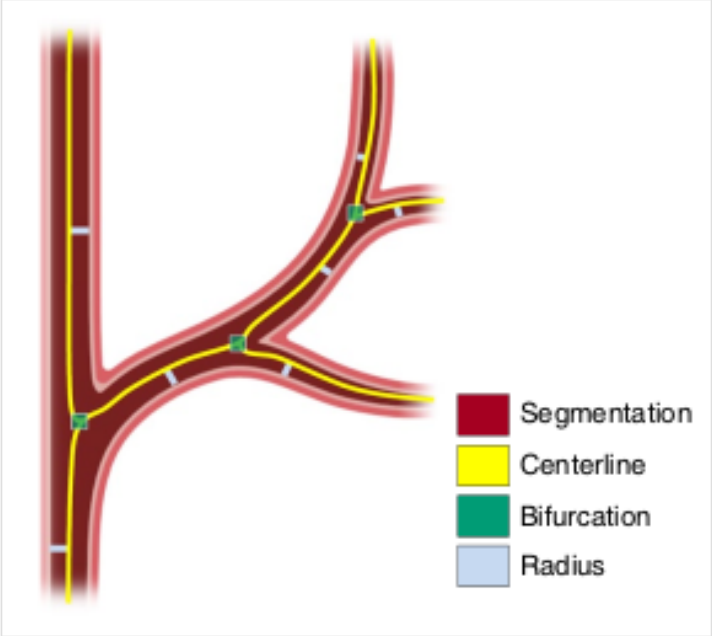


Figure 2.12: Generated annotations for the synthetic data. Segmentation covers the region contained within the vessel walls, the centerline refers to the center of the vascular tube, bifurcation is the point where the vessel split into two or more smaller vessels, and the radius is the distance from the vascular wall to the centerline.

there are available data for the task at hand, there are little or no ground truth annotations for the data. Unlike natural images,

annotating medical data require a certain level of medical expertise and time. For brain vessel analysis, the annotation process becomes even more tedious and near impossible due to the size and nature of the object of interest. To overcome this challenge, we resort to the generation of synthetic data with associated annotations. Schneider et al. [41] proposed and implemented a simulator of a vascular tree that follows a generative process inspired by the biology of angiogenesis. This same approach is developed as a complementary algorithm for filling in missing elements of a vascular tree in μCT [40]. We use the proposed simulator to generate physiologically plausible vascular trees in graph form, and transform the graph data into volumetric data that can be used in training Deep network architectures. The volumetric data generated includes annotation for vessels, centerline, vessel bifurcation, and vessel radius (see Figure 2.12). Details of the generated synthetic data, parameters used and availability can be found in Tetteh et al. [49]. The generated synthetic data is used for pre-training deep networks for vessel segmentation and as transfer learning data for centerline prediction and bifurcation detection in the work of Tetteh et al. [49].

2.5 ANALYSIS OF WHOLE MOUSE BRAIN VASCULATURE

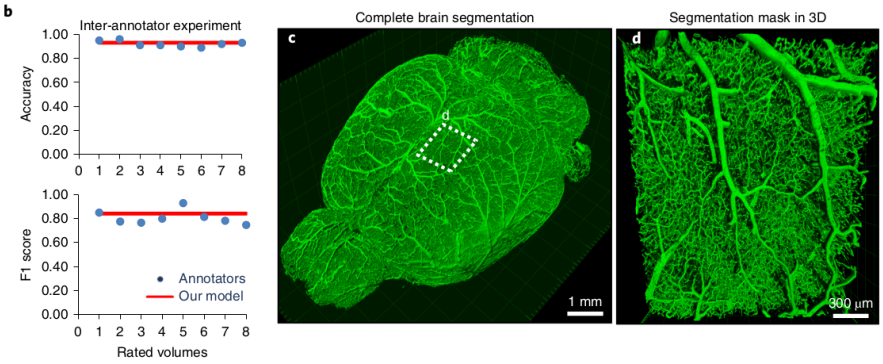


Figure 2.13: Overview of results from vessel segmentation task from the Vessel Segmentation and Analysis Pipeline (VesSAP). Left column (b), shows the accuracy and F1 score for the inter-annotator experiment (blue) as compared to VesSAP (red). Middle column (c) shows a sample of a full brain segmentation from a mouse and right column (d) shows a zoomed version of the small volume (marked white) in column (c).

Table 1 | Evaluation metrics of the different segmentation approaches for 75 volumes of 100 × 100 × 50 pixels

Segmentation model	cl-F1	Accuracy	F1 score	Jaccard	Parameters	Speed
VesSAP CNN	**0.93 ± 0.02***	**0.94 ± 0.01**	**0.84 ± 0.05**	**0.84 ± 0.04**	**0.0587 M***	**1.19 s***
VesSAP CNN, trained from scratch	**0.93 ± 0.02**	**0.94 ± 0.01**	**0.85 ± 0.04***	**0.85 ± 0.04**	**0.0587 M***	**1.19 s***
VesSAP CNN, synthetic training data	0.87 ± 0.02	0.90 ± 0.05	0.72 ± 0.07	0.70 ± 0.05	**0.0587 M***	**1.19 s***
3D U-Net	**0.93 ± 0.02**	**0.95 ± 0.01***	**0.85 ± 0.03***	**0.85 ± 0.03**	178.4537 M	61.22 s
V-Net	**0.94 ± 0.02***	**0.95 ± 0.02***	**0.86 ± 0.07***	**0.86 ± 0.07***	88.8556 M	26.87 s
Frangi vesselness	0.84 ± 0.03	0.85 ± 0.03	0.47 ± 0.19	-	-	117.00 s
Markov random field	0.86 ± 0.02	0.85 ± 0.03	0.48 ± 0.04	-	-	24.31 s

All values are given as the mean ± s.d. The best performing algorithms are in bold and highlighted with an asterisk; algorithms whose performance did not differ more than 2% from the best performing algorithms are in bold. The number of trainable parameters for deep learning architectures is given in millions (M).

Figure 2.14: A snapshot of Table 1 from Todorov et al. [52]. The segmentation model VesSAP CNN refers to the model built with our proposed cross-hair filters. A Detailed description of the other models and experimental setup can be found in Todorov et al. [52] in the Appendix.

Imaging of the complete mouse brain vasculature down to the smallest blood vessels has been made possible by recent advances in tissue clearing. However, imaging of the complete vascular network of the brain to the capillary level results in a dataset of terabyte size. Established image processing methods do not scale well with terabyte-sized image volumes due to the high memory and computational requirements. In addition, obtaining ground truth labels for such high-resolution data is tedious and requires long expert hours. As a direct application of our proposed cross-hair filter and generated synthetic data, we build a Vessel Segmentation and Analysis Pipeline (VesSAP) in the work of Todorov et al. [52]. VesSAP uses as a base the proposed computational and memory efficient cross-hair filters in a fully convolutional neural network (FCNN) architecture. The network architecture is first trained on the synthetically generated data discussed in Section 2.4 and fine-tuned on sparsely annotated mice data. Figures 2.13 and 2.14 show a sample of complete 3D segmentation from the mice data and a snapshot of quantitative results from the VesSAP experiments as presented in Todorov et al. [52]. Other results from the work (Table 1 of [52]) show that cross-hair filters used as a building block help reduce the parameter size (< 0.6 million) as compared to the state-of-the-art network architectures (> 88 million) and improve execution speed by at least 95% (1.19s vs 24.31s). Details of the image acquisition, clearing, and the experimental setup can be found in Todorov et al. [52] which is attached to the Appendix of this thesis.

Figure 2.15: A sample of patient DSA image. The red circle represents the occluded region.

2.6 COLLATERAL FLOW GRADING

Collateral circulation results from specialized anastomotic channels which are present in most tissues and capable of providing nutrient perfusion to regions with compromised blood flow due to ischemic injuries caused by ischemic stroke, coronary atherosclerosis, peripheral artery disease, and similar conditions or diseases [12]. Collateral circulation helps to sustain blood flow in the ischaemic areas and is a key factor in determining the likelihood of successful reperfusion and favorable clinical outcome [14]. MRI perfusion and diffusion have evolved as key biomarkers in determining collateralization of stroke patients, and a patient stratification based on these markers has been proposed repeatedly [43]. At the same time, a qualitative CTA and DSA-based grading are the most common approaches for evaluating collateralization [21, 22, 33]. Figure 2.15 shows a sample DSA of a patient with the occluded region marked in red.

A high number of imaging methods exist to assess the structure of the cerebral collateral circulation and several grading

criteria have been proposed to quantify the characteristics of collateral blood flow. However, this grading is mostly done through visual inspection of the acquired images which introduces two main challenges.

- First, there are *biases and inconsistencies in the current grading approaches*: There is a high tendency of introducing bias in the final grade assigned to a patient depending on the experience level of the clinician. There are inconsistencies also in the grade assigned by a particular clinician at different times for the same patient. These inconsistencies are quantified at 16% interobserver agreement and a maximum intraobserver agreement of 74% respectively in a similar study by [4].

- Second, *grading is time-consuming and tedious*: Aside from the problem of bias prediction, it also takes the clinician several minutes to go through the patient images to first select the correct image sequence, detect the region of interest and then assign a grading – a period of time which otherwise could have been invested in the treatment of the patient.

We present a set of solutions focusing on two main aspects of the task at hand. That is, the region of interest (ROI) needs to be identified, and the identified region of interest (ROI) needs to be processed and classified. We predict digitally subtracted angiography (DSA) based collateral flow grading (bad = 0, medium = 1, good = 3) from parametric MR perfusion images in this task. The dataset includes three parametric volumes for each patient. That is the time-to-peak (T_{max}) volume which represents the time taken for the blood flow to reach its peak, the relative blood flow (rBF) volume which is the volume of blood passing through a given brain tissue per unit of time, and the relative blood (rBV) volume which is defined as the volume of blood in a given brain tissue relative to an internal control (e.g. normal white matter or an arterial input function). Figure 2.16 shows samples of the three parametric volumes sampled from the three categories of collateral flow grading.

We proposed a reinforcement learning scheme for achieving the first step of identifying the ROI from the MRI perfusion volume and propose several feature extraction and classification algorithms for processing and classifying the ROI.

Results from the experiment show an acceptable level of accuracy from the region of interest identification task and suggest

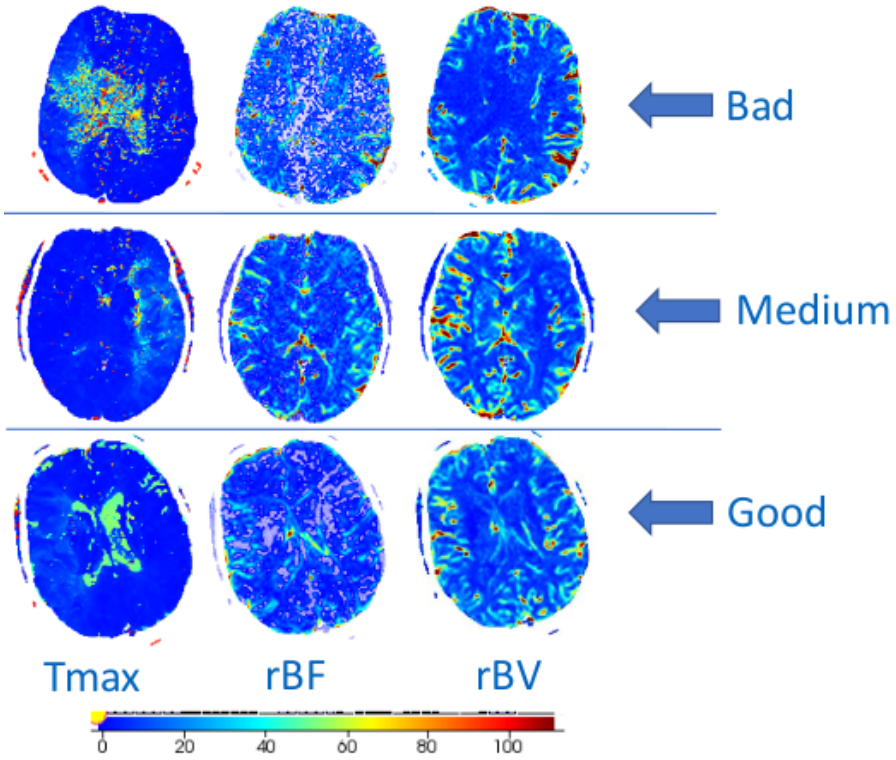


Figure 2.16: Sample of images from the MR perfusion dataset used in the collateral flow grading experiment. The Tmax column represents the time-to-peek, that is the time taken for the blood flow to reach its peak, the rBF column represents relative blood flow which is the volume of blood passing through a given brain tissue per unit of time, and the rBV column represents relative blood volume defined as the volume of blood in a given brain tissue relative to an internal control (e.g. normal white matter or an arterial input function). The upper row is a sample of slices from a patient with poor collateral flow, the middle row is from a patient with medium collateral flow and the bottom row represents a patient with good collateral flow.

that this task can be automated. The classification task also shows an overall accuracy of 72% which is very promising compared to an inter-observer agreement of 16% in a similar experiment. This is the first attempt at fully automating the collateral flow grading task and will serve as the baseline for any further research in this direction. Detailed literature review, description of the methodology used, and the intermediate and final results are presented in Tetteh et al. [50].

Part II

PUBLICATIONS

3

DEEP-FEXT: DEEP FEATURE EXTRACTION FOR VESSEL SEGMENTATION AND CENTERLINE PREDICTION

Authors: **Giles Tetteh**, Markus Rempfler, Claus Zimmer, and Bjoern H. Menze

Abstract: Feature extraction is a very crucial task in image and pixel (voxel) classification and regression in biomedical image modelling. In this work we present a feature extraction scheme based on inception models for pixel classification tasks. We extract features under multi- scale and multi-layer schemes through convolutional operators. Layers of Fully Convolutional Network are later stacked on these feature extraction layers and trained end-to-end for the purpose of classification. We test our model on the DRIVE and STARE public data sets for the purpose of segmentation and centerline detection and it outperforms most existing hand crafted or deterministic feature schemes found in literature. We achieve an average maximum Dice of 0.85 on the DRIVE data set which outperforms the scores from the second human annotator of this data set. We also achieve an average maximum Dice of 0.85 and kappa of 0.84 on the STARE data set. Even though these datasets are only 2-D we also propose ways of extending this feature extraction scheme to handle 3-D datasets.

Date of publication: 10 September 2017
Conference name: 8th International Workshop, MLMI 2017 Held in Conjunction with MICCAI 2017 Quebec City, QC, Canada (Peer-reviewed)

Contribution of thesis author: project conception and coordination, experimental design, algorithm development and implementation, data analysis, leading author of manuscript.

This article has been included in "manuscript version" instead of "published version".

Deep-FExt: Deep Feature Extraction for Vessel Segmentation and Centerline Prediction

Giles Tetteh[2], Markus Rempfler[2], Claus Zimmer[1], Bjoern H. Menze[2,3]

[1] Neuroradiology, Klinikum Rechts der Isar, TU München, Munich, Germany
[2] Department of Computer Science, TU München, Munich, Germany
[3] Institute for Advanced Study, TU München, Munich, Germany

Abstract. Feature extraction is a very crucial task in image and pixel (voxel) classification and regression in biomedical image modelling. In this work we present a feature extraction scheme based on inception models for pixel classification tasks. We extract features under multi-scale and multi-layer schemes through convolutional operators. Layers of Fully Convolutional Network are later stacked on these feature extraction layers and trained end-to-end for the purpose of classification. We test our model on the DRIVE and STARE public data sets for the purpose of segmentation and centerline detection and it outperforms most existing hand crafted or deterministic feature schemes found in literature. We achieve an average maximum Dice of 0.85 on the DRIVE data set which outperforms the scores from the second human annotator of this data set. We also achieve an average maximum Dice of 0.85 and kappa of 0.84 on the STARE data set. Even though these datasets are only 2-D we also propose ways of extending this feature extraction scheme to handle 3-D datasets.

Keywords: Feature extraction, image and pixel classification and regression, biomedical image modelling, inception models, convolutional networks, vessel segmentation, centerline prediction.

1 Introduction

Most recent research in biomedical modelling involves qualitative and quantitative classification of a single pixel (voxel), a region of interest ROI and or an image (volume). These classification tasks mostly involve three main steps: feature extraction, feature selection and classification [1]. Out of these three steps, the feature extraction step is the most crucial since it determines which information will be present or discarded in the next steps.

Feature extraction is the process of generating features to be used in the selection and classification tasks[1]. In whole image or volume classification, feature extraction and selection can serve as a dimensionality reduction where a subset of the extracted features is selected to eliminate redundant features while maintaining the underlying discriminatory information[2]. The newly extracted

features are normally of lower dimension than the original feature space. However, most pixelwise feature extraction tasks lead to dimensionality extension. That is, a new set of features of high dimension is extracted for each given pixel based on its neighbourhood.

Feature extraction techniques come mainly in three main flavours - hand crafted texture features, supervised learned features and unsupervised feature extraction.

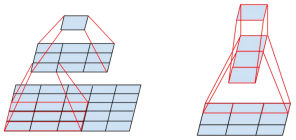
Textures are complex visual patterns composed of entities, or subpatterns, that have characteristic brightness, colour, slope, or size [3]. The local subpattern properties give rise to the perceived lightness, uniformity, density, roughness, regularity, linearity, frequency, phase, directionality, coarseness, randomness, fineness, smoothness, or granulation of the texture as a whole [4]. For a review of texture features, categorization and various uses one can refer to [3].

Other groups of hand crafted features are based on differential geometry and the analysis of gradient and Hessian of pixel intensity. These are mostly used as image enhancement to objects of specific shape of interest in a given image. For example in [5] the multiscale second order local structure of an image (Hessian) is examined with the purpose of developing a vessel enhancement filter. Ultimately, a vesselness measure is obtained on the basis of the eigenvalues of the Hessian. This vesselness measure serves as a measure of the likelihood of the presence of geometrical structures which can be regarded as tubular. Also a curvilinear structure detector, called Optimally Oriented Flux (OOF) finds an optimal axis on which image gradients are projected in order to compute the image gradient flux[6].

The second class of feature extraction techniques are in the form of unsupervised learning and transfer learning. These are mainly autoencoders and its variations like restricted Boltzmann's machine. Autoencoders are simple learning circuits which aim to transform inputs into outputs with the least possible amount of distortion [7]. For detailed discussion of autoencoders, unsupervised learning and deep architectures one can refer to [7]. These architectures though very simple are very important in the field of machine learning and form the base components of deep learning architectures.

Architectures like CNN and other deep networks also extract hierarchical features in a supervised manner through the use of ground truth annotations. Szegedy et al. [8] proposed the inceptions model as a way of building deeper networks capable of learning and extracting dense feature while maintaining acceptable speed and memory usage. This idea has been used in building the GoogLeNET [8] which achieves the state of the art results on image classification tasks.

In this paper we discuss briefly inception models in general and extend the idea to build feature extraction layers in an autoencoder fashion. We will also discuss how to stack these pixelwise feature extraction layers to form a deep architecture which is then fine-tuned for the purpose of supervised learning.



(i) (ii)

Fig.1: (i): Mini-network replacing a 5x5 convolutional operation. (ii): Mini-network replacing a 3x3 convolutional operation

2 Methodology

2.1 Inception Models

The main idea of the Inception architecture is based on finding out how an optimal local sparse structure in a convolutional vision network can be approximated and covered by readily available dense components [8]. Inception based networks replaces convolutional operations with mini-networks which uses less parameters and less computation. A convolution with a filter size of 5×5 can be replaced with a mini-network of two layers of filter sizes 3×3 each as shown in Figure 1i. This reduces the parameter size from 25 (i.e. 5×5) to 18 (i.e. $3 \times 3 + 3 \times 3$ Similarly a convolutional operation with filter size 3×3 can be replaced with a mini-network of two layers with filters 1×3 and 3×1 respectively as shown in Figure 1ii.

By factorizing convolutional operations with bigger filter sizes into mini-networks with smaller filter sizes [9] proposed building a network which make use of filters with sizes not greater than 3×3. This helps to conserve memory and computational time which can be used to increase the depth of the network to improve performance. Inception modules as described in [8,9] form the building layers of the state of the art GoogLeNet network which was presented to the ILSVRC14 competition. Thorough discussion of inception architure can be found in [8,9]. The original inception models are used in networks meant for full image classification. In the next section we discuss adapting the inception model to form a feature extraction layer in pixel wise classification tasks.

2.2 Pixelwise feature extraction layer

The original inception architecture described in Section 2.1 is designed to fit in the domain of full image classification. This therefore leads to feature or dimensionality reduction. However, in this section we are rather interested in extracting features for pixel classification. In order to achieve this aim we first take the following two steps:

1. Remove all pooling operations. Pooling operations are used in image based classification tasks to extract invariant features and to reduce the dimension in the downstream layers of the network. Pooling layers work by replacing a region of an image by a statistic (e.g. mean, or maximum) of that region. This helps in image based classification by removing noise and outliers. However, pooling leads to loss of fine local details which is very crucial in pixelwise (voxelwise)

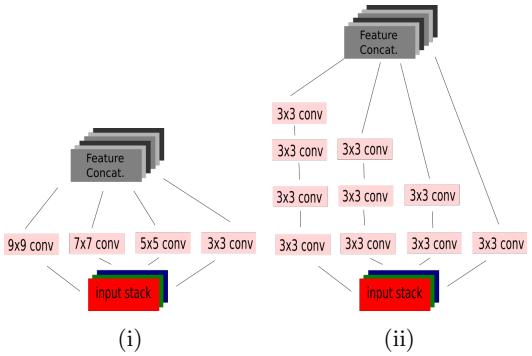


Fig. 2: Feature extraction layers: figure i represents a layer without network factorization and figure ii represents a layer after network factorization.

tasks. This is a major problem in applying deep learning to detection problems in medical images. We therefore remove all pooling layers to make our feature extraction layers robust to object of interest of all sizes. Again we note that this is done only in the feature extraction layers not in the classifier which is later stack on the extracted features.
2. All convolution operations result in an output of the same size as the input. Here the idea is to keep the feature extraction layers as simple as possible such that we can stack them. We do this by padding the input to each layer with enough zeros so that it takes care of the number of pixels or voxels that are lost through the convolutional operation. This makes stacking the layers easier without thinking about the output shape of the previous layers.

we then choose a set of scales s e.g. $\{3, 5, 7, 9\}$ design a multiscale layer as shown in Figure 2i. We then replace all convolutions with bigger than 3×3 filter sizes with mini-networks as described in Section 2.1 to obtain our final feature extraction layer in Figure 2ii. By stacking multiple layers together we build a feature extraction network suitable for pixel classification. We note that the output from each layer is further transformed by a non-linear activation function (rectified linear units - ReLU) before it moves to the next layer. The concatenated output from each layer together with the input image are further concatenated to form the final feature set as shown in Figure 3. We refer to this deep feature extraction network as Deep-FExt in the rest of the paper.

3 Experiments

To test Deep-FExt we design a network of 5 feature extraction layers which extract a total of 100 features per pixel (see Figure 1). We then create a 10×10 feature mesh from each pixel feature set. Hence each pixel is then represented by 2-D image of size 10×10. We first train the feature extraction network in an unsupervised manner and then stack a CNN with 3 layers and randomly initialized parameters on the feature mesh and fine-tune end-to-end using stochastic gradient descent for classification and prediction purposes.

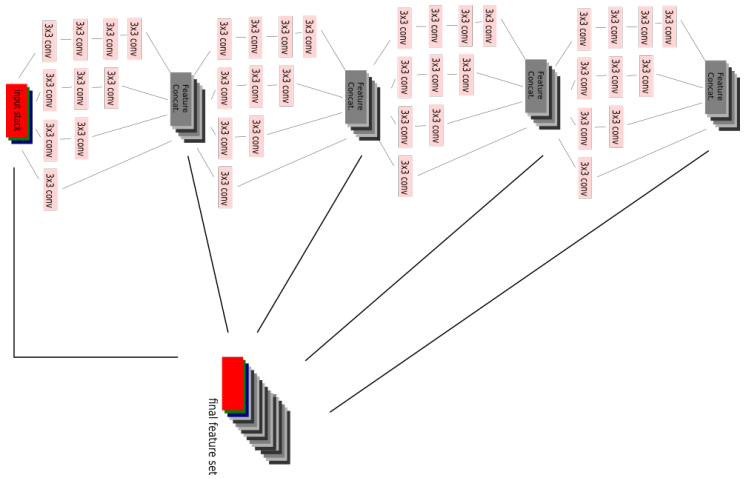


Fig. 3: Feature extraction network with final feature set from multiple layers

Table 1: Feature extraction network structure employed in our experiment.

Layer	Input type and size	Filter sizes (extracted feats)	Total features
1	RGB image with 3 channels	3(5),5(5),7(5),9(3),11(3)	21
2	concat features from layer 1	3(5),5(5),7(5),9(3),11(3)	21
3	concat features from layer 2	3(5),5(4),7(4),9(3),11(3)	19
4	concat features from layer 3	3(4),5(4),7(4),9(3),11(3)	18
5	concat features from layer 4	3(4),5(4),7(4),9(3),11(3)	18
	Total	97 + 3 (input RGB) = 100	

The full network structure is described in Table 1. Qualitative visualizations in Figure 4 show that Deep-FExt is able to extract hierarchical features ranging from edge detectors, intensity gradients, and curvature at different scales.

3.1 Vessel Segmentation

For vessel segmentation we experiment on the DRIVE [10] and STARE [11] datasets. The DRIVE dataset is made up of 20 training examples and 20 test examples with two annotations in each group. We use the first annotation as the ground truth for training our network and testing. We also compare our results to the second annotation. The STARE dataset is made up of 20 annotated images with two annotations each. We split the data into 10 images for training and the remaining 10 for testing. Our results (See Table 2) show that our Deep-FExt network outperforms most of the existing architecture on the segmentation of the DRIVE and STARE datasets. Results for Deep Retinal Understanding (DRIU) [12] are obtained by evaluating pre-computed probability maps provided on the

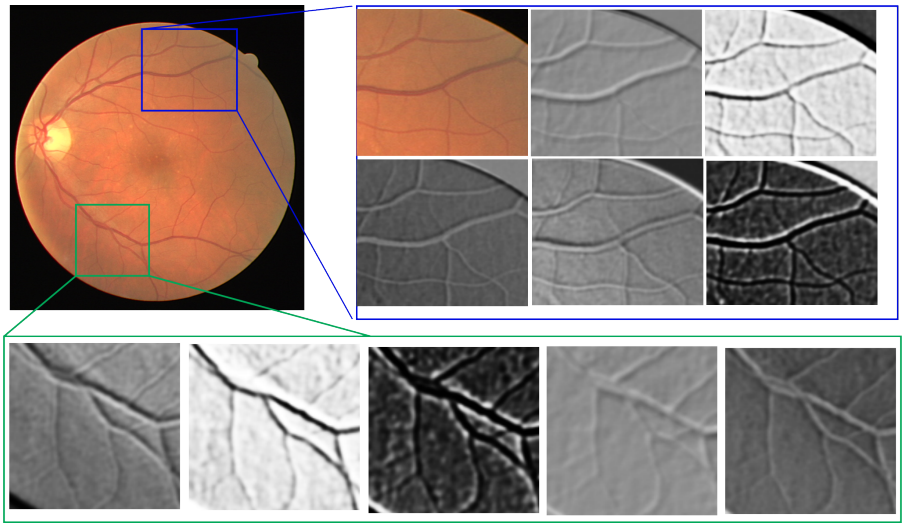


Fig. 4: Actual image (top left). We observe that Deep-FExt is able to learn features that resemble edge detectors, intensity gradients, and curvature at different scales.

paper's page. DRIU outperforms Deep-FExt in dice however, DRIU uses VGG as a base network, which is much deeper, carefully pre-trained and fine-tuned on millions of images. In contrast, Deep-FExt uses a simple SGD and does not employ any pre-training, or special parameter initialization. Yet we achieved results which is less than 3% lower than DRIU in dice. Other results are also stated as reported by [12].

3.2 Centerline Prediction

We again test Deep-FExt on DRIVE and STARE datasets for the purpose of centerline prediction. We generated centerline annotations by applying skeletonization to the the various manual annotations and used the same training and testing splits that were used for the vessel segmentation. We evaluated our results based on centerline prediction alone (OC) and a combined multi-class prediction of centerline and vessel (B). We compare our results to the second annotator of these datasets (see Table 3).

4 Conclusion

Deep-FExt outperforms most of the existing architectures on the DRIVE and STARE datasets. We believe Deep-FExt can be used to extract feature for general medical image segmentation tasks. By replacing the 2-D convolutions with 3-D we can also extend Deep-FExt to handle medical volumes. With the idea of

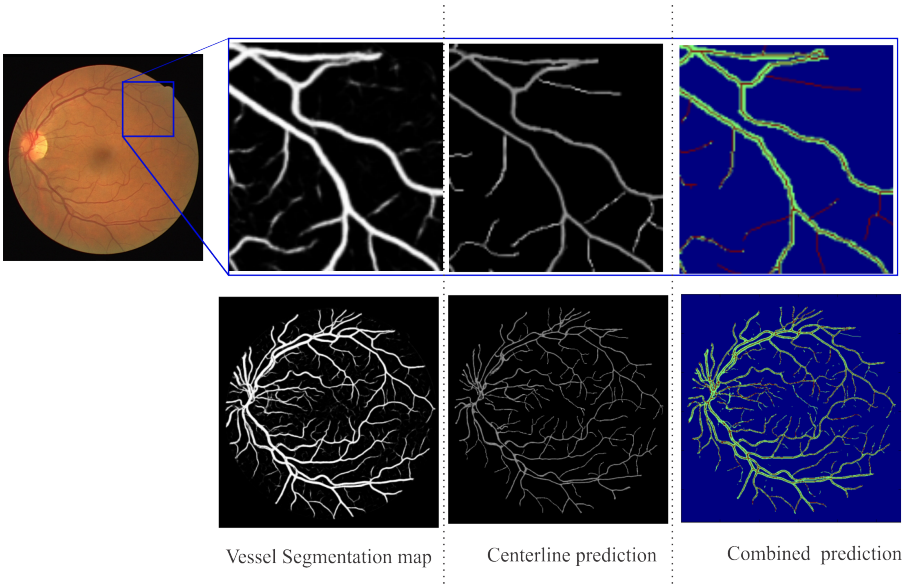


Fig. 5: Qualitative predictions from Deep-FExt. Top images show close view of region marked blue in the original image.

Table 2: Vessel segmentation results on the DRIVE and STARE datasets. AMD refers to the average maximum Dice.

Dataset	Method	Precision	Recall	Dice	AMD	Kappa
DRIVE	Deep-FExt	80.44	80.32	80.38	84.67	78.48
	DRIU [12]	81.59	82.61	82.10	86.02	80.34
	N^4 fields [13]			80.50		
	Kernel Boost[14]			80.00		
	HED [15]			79.60		
	CRFs [16]			78.10		
	2nd Annotator	80.40	77.46	78.90	82.98	76.90
STARE	Deep-FExt	82.04	79.54	80.78	84.87	79.20
	DRIU [12]	82.67	83.80	83.23	86.28	81.84
	HED [15]			80.50		
	2nd Annotator	63.65	94.46	76.05	79.66	73.64

mini-networks memory is conserved and speed is also improved. As further research, we consider experimenting with training Deep-FExt in an unsupervised manner similar to autoencoders. This would be valuable for generating features for clustering, or in situations where supervised learning is not feasible due to lack of annotated data.

Table 3: Centerline prediction results on the DRIVE and STARE datasets. Metrics are computed on a pixel level. AMD refers to the average maximum Dice.

Dataset	Method	Precision	Recall	Dice	AMD	Kappa
DRIVE	Deep-FExt(OC)	57.95	82.04	67.92	72.30	66.88
	Deep-FExt(B)	71.38	74.65	72.98	77.20	71.49
	2nd Annotator(OC)	60.38	45.86	52.13	63.95	44.72
	2nd Annotator(B)	70.45	69.35	69.89	69.89	67.31
STARE	Deep-FExt(OC)	53.63	74.27	62.29	75.98	61.45
	Deep-FExt(B)	73.33	75.73	74.51	79.72	72.45
	2nd Annotator(OC)	57.51	52.43	54.85	66.42	40.72
	2nd Annotator(B)	63.23	75.99	69.02	72.27	65.54

References

1. Choras, R.S.: Image feature extraction techniques and their applications for cbir and biometrics systems. International Journal of Biology and Biomedical Engineering 1, 6–15 (2007)
2. Adegoke, B.O., Ola, B.O., Omotayo, M.E.: Review of feature selection methods in medical image processing. IOSR Journal of Engineering (IOSRJEN) 4, 1 (2014)
3. Materka, A., Strzelecki, M.: Texture analysis methods a review. University of Lodz, Institute of Electronics, COST B11 report (1998)
4. Levine, M.: Vision in Man and Machine. McGraw-Hill (1985)
5. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. MICCAI (1998)
6. Law, M.W.K., Chung, A.C.S.: Three dimensional curvilinear structure detection using optimally oriented flux ECCV 2008, pp. 368–382.
7. Baldi, P.: Autoencoders, unsupervised learning, and deep architectures. Workshop on Unsupervised and Transfer Learning. JMLR: Workshop and Conference Proceedings 50 (2012)
8. Szegedy, C., et al.: Going deeper with convolutions. CVPR (2015)
9. Szegedy, C., et al.: Rethinking the inception architecture for computer vision. CVPR (2016)
10. Staal, J., et al.: Ridge-based vessel segmentation in color images of the retina. IEEE T-MI 23(4) p. 501509 (2004)
11. Hoover, A., Kouznetsova, V.and Goldbaum, M.: Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. IEEE T-MI 19(3) p. 203210 (2000)
12. Maninis, K.K., Pont-Tuset, J., Arbeláez, P., L, V.G.: Deep retinal image understanding. MICCAI 9901 (2016)
13. Ganin, Y., Lempitsky, V.: N4-fields: Neural network nearest neighbor fields for image transforms. ACCV (2014)
14. Becker, C., Rigamonti, R., Lepetit, V., Fua, P.: Supervised feature learning for curvilinear structure segmentation. MICCAI (2013)
15. Xie, S., Tu, Z.: Holistically-nested edge detection. ICCV pp. 142–154 (2015)
16. Orlando, J., Blaschko, M.: Learning fully-connected crfs for blood vessel segmentation in retinal images. MICCAI (2014)

4

DEEPVESSELNET: VESSEL SEGMENTATION, CENTERLINE PREDICTION AND BIFURCATION DETECTION IN 3-D ANGIOGRAPHIC VOLUMES

Authors: Giles Tetteh, Velizar Efremov, Nils D. Forkert, Matthias Schneider, Jan Kirschke, Bruno Weber, Claus Zimmer, Marie Piraud and Björn H. Menze

Abstract: We present DeepVesselNet, an architecture tailored to the challenges faced when extracting vessel trees and networks and corresponding features in 3-D angiographic volumes using deep learning. We discuss the problems of low execution speed and high memory requirements associated with full 3-D networks, high-class imbalance arising from the low percentage (<3%) of vessel voxels, and unavailability of accurately annotated 3-D training data – and offer solutions as the building blocks of DeepVesselNet. First, we formulate 2-D orthogonal cross-hair filters which make use of 3-D context information at a reduced computational burden. Second, we introduce a class balancing cross-entropy loss function with false-positive rate correction to handle the high-class imbalance and high false positive rate problems associated with existing loss functions. Finally, we generate a synthetic dataset using a computational angiogenesis model capable of simulating vascular tree growth under physiological constraints on local network structure and topology and use these data for transfer learning. We demonstrate the performance on a range of angiographic volumes at different spatial scales including clinical MRA data of the human brain, as well as CTA microscopy scans of the rat brain. Our results show that cross-hair filters achieve over 23% improvement in speed, lower memory footprint, lower network complexity which prevents overfitting and comparable accuracy that does not differ from full 3-D filters. Our class balancing metric is crucial for training the network, and transfer learning with synthetic data is an efficient, robust, and very generalizable approach leading to a network that excels in a variety of angiography segmentation tasks. We observe that sub-sampling and max pooling layers may lead to a drop in performance in tasks

that involve voxel-sized structures. To this end, the DeepVessel-Net architecture does not use any form of sub-sampling layer and works well for vessel segmentation, centerline prediction, and bifurcation detection. We make our synthetic training data publicly available, fostering future research, and serving as one of the first public datasets for brain vessel tree segmentation and analysis.

Date of publication: 08 December 2020
Journal: Frontiers in Neuroscience

Contribution of thesis author: project initialization, methodology development, algorithm development and testing, results analysis and lead author of paper.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 08 December 2020
doi: 10.3389/fnins.2020.592352

DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes

Giles Tetteh[1]*, *Velizar Efremov*[1,2], *Nils D. Forkert*[3], *Matthias Schneider*[2], *Jan Kirschke*[4], *Bruno Weber*[2], *Claus Zimmer*[4], *Marie Piraud*[1] and *Björn H. Menze*[1,5]

[1] Department of Computer Science, TU München, München, Germany, [2] Institute of Pharmacology and Toxicology, University of Zurich, Zurich, Switzerland, [3] Department of Radiology, University of Calgary, Calgary, AB, Canada, [4] Neuroradiology, Klinikum Rechts der Isar, TU München, München, Germany, [5] Department for Quantitative Biomedicine, University of Zurich, Zurich, Switzerland

We present DeepVesselNet, an architecture tailored to the challenges faced when extracting vessel trees and networks and corresponding features in 3-D angiographic volumes using deep learning. We discuss the problems of low execution speed and high memory requirements associated with full 3-D networks, high-class imbalance arising from the low percentage (<3%) of vessel voxels, and unavailability of accurately annotated 3-D training data—and offer solutions as the building blocks of DeepVesselNet. First, we formulate 2-D orthogonal cross-hair filters which make use of 3-D context information at a reduced computational burden. Second, we introduce a class balancing cross-entropy loss function with false-positive rate correction to handle the high-class imbalance and high false positive rate problems associated with existing loss functions. Finally, we generate a synthetic dataset using a computational angiogenesis model capable of simulating vascular tree growth under physiological constraints on local network structure and topology and use these data for transfer learning. We demonstrate the performance on a range of angiographic volumes at different spatial scales including clinical MRA data of the human brain, as well as CTA microscopy scans of the rat brain. Our results show that cross-hair filters achieve over 23% improvement in speed, lower memory footprint, lower network complexity which prevents overfitting and comparable accuracy that does not differ from full 3-D filters. Our class balancing metric is crucial for training the network, and transfer learning with synthetic data is an efficient, robust, and very generalizable approach leading to a network that excels in a variety of angiography segmentation tasks. We observe that sub-sampling and max pooling layers may lead to a drop in performance in tasks that involve voxel-sized structures. To this end, the DeepVesselNet architecture does not use any form of sub-sampling layer and works well for vessel segmentation, centerline prediction, and bifurcation detection. We make our synthetic training data publicly available, fostering future research, and serving as one of the first public datasets for brain vessel tree segmentation and analysis.

Keywords: vascular network, cross-hair filters, deepvesselnet, bifurcation, vessel segmentation, centerline, class balancing, vascular tree

OPEN ACCESS

Edited by:
Ayman Sabry El-Baz,
University of Louisville, United States

Reviewed by:
Karl Helmer,
Massachusetts General Hospital and Harvard Medical School, United States
Shijie Zhao,
Northwestern Polytechnical University, China

***Correspondence:**
Giles Tetteh
giles.tetteh@tum.de

Specialty section:
This article was submitted to Brain Imaging Methods, a section of the journal Frontiers in Neuroscience

Received: *06 August 2020*
Accepted: *16 November 2020*
Published: *08 December 2020*

Citation:
Tetteh G, Efremov V, Forkert ND, Schneider M, Kirschke J, Weber B, Zimmer C, Piraud M and Menze BH (2020) DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes. Front. Neurosci. 14:592352. doi: 10.3389/fnins.2020.592352

1. INTRODUCTION

Angiography offers insights into blood flow and conditions of the vascular tree. Three dimensional volumetric angiography information can be obtained using magnetic resonance (MRA), ultrasound, or x-ray based technologies like computed tomography (CT). A common first step in analyzing these data is vessel segmentation. Still, moving from raw angiography images to vessel segmentation alone might not provide enough information for clinical use, and other vessel features like centerline, diameter, or bifurcations of the vessels are also needed to accurately extract information about the vascular tree, for example, to characterize its structural properties or flow pattern. In this work, we present a deep learning approach, called DeepVesselNet, to perform vessel segmentation, centerline prediction, and bifurcation detection tasks. We make the code available (Tetteh, 2019a), and a ready-to-use implementation is available as companion material to our study "Machine learning analysis of whole mouse brain vasculature" (Todorov et al., 2020). DeepVesselNet deals with challenges that result from speed and memory requirements, unbalanced class labels, and the difficulty of obtaining well-annotated data for curvilinear volumetric structures by addressing the following three key limitations.

Processing 3-D medical volumes poses a memory consumption and speed challenge. Using 3-D convolutional neural networks (CNNs) leads to drastic increase in number of parameters to be optimized and computations to be executed when compared to 2-D CNNs. At the same time, applying a 2-D CNN in a slice-wise fashion discards valuable 3-D context information that is crucial for tracking curvilinear structures in 3-D. Inspired by the ideas of Rigamonti et al. (2013), Roth et al. (2014), and Liu et al. (2017) who proposed separable filters and used intersecting 2-D planes, we demonstrate the use of cross-hair filters from three intersecting 2-D filters, which helps to avoid the memory and speed problems of classical 3-D networks, while at the same time making use of 3-D information in volumetric data. Unlike the existing ideas where 2-D planes are extracted at a pre-processing stage and used as input channels (see discussion in section 2.1.2), our cross-hair filters are implemented on a layer level which help to retain the 3-D information throughout the network (see section 2.1).

The vessel, centerline and bifurcation prediction tasks is characterized by high class imbalances. Vessels account for $<3\%$ of the total voxels in a patient volume, centerlines represent a fraction of the segmented vessels, and visible bifurcations are in the hundreds at best—even when dealing with volumes with 10^6 and more voxels. This bias toward the background class is a common problem in medical data (Grzymala-Busse et al., 2004; Christ et al., 2016; Haixiang et al., 2017). Unfortunately, current class balancing loss functions for training CNNs turn out to be numerically unstable in extreme cases as ours. We offer a solution to this "extreme class imbalance" problem by introducing a new loss function (see section 2.2) that we demonstrate to work well with our vascular features of interest.

Manually annotating vessels, centerlines, and bifurcations requires many hours of work and expertise. To this end, we demonstrate the successful use of simulation based frameworks (Szczerba and Székely, 2005; Schneider et al., 2012, 2014) that can be used for generating synthetic data with accurate labels (see section 2.3) for pre-training our networks, rendering the training of our supervised classification algorithm feasible. The transfer learning approach turns out to be a critical component for training CNNs in a wide range of angiography tasks and applications ranging from CT micrographs to TOF MRA. The synthesized and the clinical MRA datasets are made available publicly for future research and validation purposes. Further description and download link is provided in section 3.1.

1.1. Prior Work and Open Challenges

1.1.1. Vessel Segmentation

Vessel enhancement and segmentation is a longstanding task in medical image analysis (see reviews by Kirbas and Quek, 2004; Lesage et al., 2009). The range of methods employed for vessel segmentation reflect the development of image processing during the past decades, including region growing techniques (Martínez-Pérez et al., 1999), active contours (Nain et al., 2004), statistical and shape models (Chung and Noble, 1999; Young et al., 2001; Liao et al., 2013; Moreno et al., 2013), particle filtering (Florin et al., 2006; Wörz et al., 2009; Dalca et al., 2011), and path tracing (Wang et al., 2013). All of these examples are interactive, starting from a set of seed labels as root and propagating toward the branches. Other approaches aim at an unsupervised enhancement of vascular structures: a popular multi-scale second order local structure of an image (Hessian) was examined by Frangi et al. (1998) with the purpose of developing a vessel enhancement filter. A measure of vessel-likeliness is then obtained as a function of all eigenvalues of the Hessian. A novel curvilinear structure detector, called Optimally Oriented Flux (OOF) was proposed by Law and Chung (2008) to find an optimal axis on which image gradients are projected to compute the image gradient flux. OOF has a lower computational load than the calculation of the Hessian matrix proposed in Frangi et al. (1998). A level-set segmentation approach with vesselness-dependent anisotropic energy weights is presented and evaluated in Forkert et al. (2013, 2011) for 3-D time-of-flight (TOF) MRA. Phellan and Forkert (2017) presented a comparative analysis of the accuracy gains in vessel segmentation generated by the use of nine vessel enhancement algorithms on time-of-flight MRA that included multi scale vesselness algorithms, diffusion-based filters, and filters that enhance tubular shapes and concluded that vessel enhancement algorithms do not always lead to more accurate segmentation results compared to segmenting non-enhanced images directly. An early machine learning approach for vessel segmentation was proposed by Schneider et al. (2015), combining joint 3-D vessel segmentation and centerline extraction using oblique Hough forest with steerable filters. In a similar fashion, Ciresan et al. (2012) used deep artificial neural network as a pixel classifier to automatically segment neuronal structures in stacks of electron microscopy images, a task somewhat similar to vessel segmentation. One example using deep learning architecture is the work of Phellan et al. (2017) who used a deep convolutional neural network to automatically segment the vessels of the brain in TOF MRA by extracting manually annotated bi-dimensional image patches in the axial, coronal, and sagittal directions as an

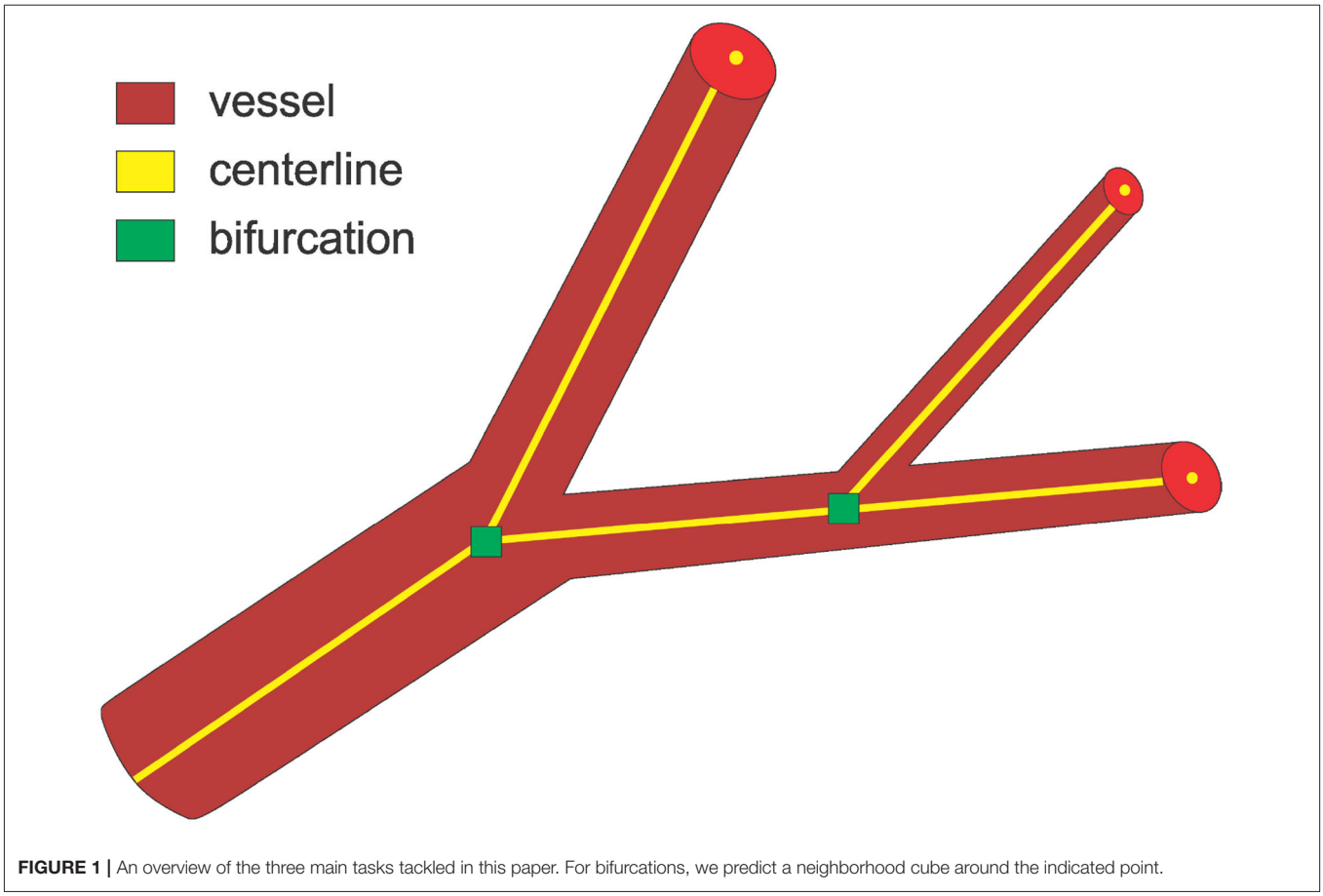


FIGURE 1 | An overview of the three main tasks tackled in this paper. For bifurcations, we predict a neighborhood cube around the indicated point.

input to the training process. Koziński et al. (2018) proposed a loss function that accommodates ground truth annotations of 2-D projections of the training volumes, for training deep neural networks in tasks where obtaining full 3-D annotations is a challenge.

Though deep learning has been applied in many medical imaging tasks, there are no dedicated architectures so far for vessel segmentation in 3-D volumetric datasets. Existing architectures might be sub-optimal and not work directly out of the box due to the unique nature of the vasculature as compared to other imaging tasks. There is therefore the need to explore other architectures and training strategies.

1.1.2. Centerline Prediction

Identifying the center of a vessel is relevant for calculating the vessel diameter, but also for obtaining the "skeleton" of a vessel when extracting the vascular tree or network (see **Figure 1**). The vessels' skeleton and center can be found by post-processing a previously generated vessel segmentation. A method based on morphological operations is developed by Shagufta et al. (2014) which performs erosion using 2×2 neighborhoods of a pixel to determine if a pixel is a centerline candidate. Active contour models are applied in Maddah et al. (2003) as well as path planning and distance transforms for extracting centerline in vessels, and Chen and Cohen (2015) proposed a geodesic or minimal path technique. A morphology-guided level set model is used in Santamaría-Pang et al. (2007) to performed centerline extraction by learning the structural patterns of a tubular-like object, and estimating the centerline of a tubular object as the path with minimal cost with respect to outward flux in gray level images. Vesselness filters were adopted by Zheng et al. (2012) to predict the location of the centerline, while Macedo et al. (2010) used Hough transforms in handling a similar task. A Hough random forest with local image filters is designed in Schneider et al. (2015, 2012) to predict the centerline, and trained on centerline data previously extracted using one of the level set approaches.

The application of deep learning to the extraction of vessel centerline has not been explored. One reason may be the lack of annotated data necessary to train deep architectures that is hard to obtain especially in 3-D datasets.

1.1.3. Bifurcation Detection

A vessel bifurcation refers to the point on a vessel centerline where the vessel splits into two vessels (see **Figure 1**). Bifurcations represent the nodes of the vascular tree or network and knowing their locations is important both for network extraction and for studying its properties (Rempfler et al., 2015). They represent structures that can easily be used as landmarks in image registration, but also indicate the locations of modified blood

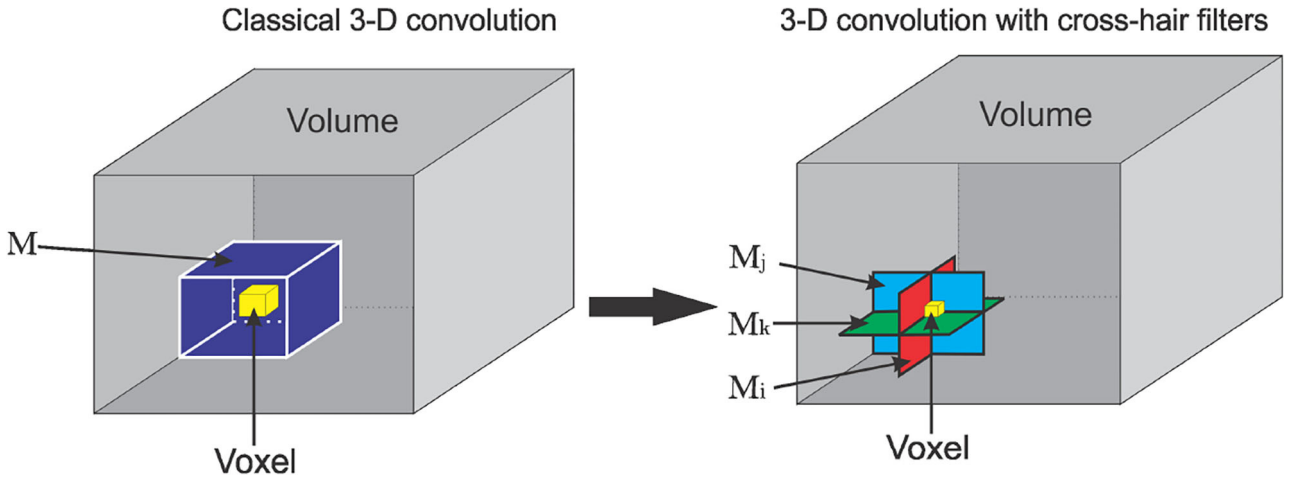


FIGURE 2 | Graphical representation of cross-hair filters for 3-D convolutional operation. **(Left)** A classical 3-D convolution with filter M. **(Right)** Cross-hair 3-D convolutional with 2-D filter stack M_i, M_j, M_k.

flow velocity and pressure within the network itself (Chaichana et al., 2017). Bifurcations are hard to detect in volumetric data as they are rare point-like features that vary in size and shape significantly. Similar to centerline extraction, the detection of bifurcations often happens by post-processing a previously generated vessel segmentation or by searching a previously extracted vessel graph. A two staged deep learning architecture is proposed in Zheng et al. (2015) for detecting carotid artery bifurcations as a specific landmark in volumetric CT data by first training a shallow network for predicting candidate regions followed by a sparse deep network for final prediction. A three stage algorithm for bifurcation detection is proposed in Chaichana et al. (2017) for digital eye fundus images, a 2-D task, and their approach included image enhancement, clustering, and searching the graph for bifurcations.

The direct predicting of the location of centerlines and bifurcations without a previous segmentation of vessels as an intermediate step is a task which has not been attempted yet. We foresee that having directly predicted centerlines and bifurcations together with those from postprocessing vessel segmentations will enhance the overall robustness and accuracy of the analysis of angiographic volumes.

2. METHODOLOGY

In the design of our DeepVesselNet architecture, we offer three methodological contributions: A. introducing fast cross-hair filters, B. dealing with extreme class balancing by relying on a loss function with stable weights, and C. generating synthetic 3D vessel structures for training DeepVesselNet and other standard segmentation architectures.

2.1. Cross-Hair Filters Formulation

In this section, we introduce the 3-D convolutional operator, which utilizes cross-hair filters to improve speed and memory usage while maintaining accuracy. For a graphical representation of classical 3-D convolutional operator and the proposed cross-hair filters is see **Figure 2**. Let I be a 3-D volume, M a 3-D convolutional kernel of shape (k_x, k_y, k_z), and $*$ be a convolutional operator. We define $*$ as:

$$I * M = A = \{a_{ijk}\};\ a_{ijk} = \sum_{r=1}^{k_x}\sum_{s=1}^{k_y}\sum_{t=1}^{k_z} I_{(R,S,T)} M_{(r,s,t)}; \quad (1)$$

$$R = i + r - \left(1 + \left[\frac{k_x}{2}\right]\right), \quad (2)$$

$$S = j + s - \left(1 + \left[\frac{k_y}{2}\right]\right), \quad (3)$$

$$T = k + t - \left(1 + \left[\frac{k_z}{2}\right]\right),$$

where $\{a_{ijk}\}$ is a position element of matrix A, $I_{(R,S,T)}$ is the intensity value of image I at voxel position (R, S, T), $M_{(r,s,t)}$ is the value of kernel M at position (r, s, t), and $[c]$ is the greatest integer less or equal to c.

From Equation (1), we see that a classical 3-D convolution involves $k_x k_y k_z$ multiplications and $k_x k_y k_z - 1$ additions for each voxel of the resulting image. For a $3 \times 3 \times 3$ kernel, we have 27 multiplications and 26 additions per voxel. Changing the kernel size to $5 \times 5 \times 5$ increases the complexity to 125 multiplications and 124 additions per voxel. This then scales up with the dimension of the input image. For example, a volume of size $128 \times 128 \times 128$ and a $5 \times 5 \times 5$ kernel results in about 262×10^6 multiplications and 260×10^6 additions. To handle this increased computational complexity, we approximate the standard 3-D convolution operation by

$$a_{ijk} = \sum_{s=1}^{k_y}\sum_{t=1}^{k_z} I_{(i,S,T)} M^i_{(s,t)} + \sum_{r=1}^{k_x}\sum_{t=1}^{k_z} I_{(R,j,T)} M^j_{(r,t)} + \sum_{r=1}^{k_x}\sum_{s=1}^{k_y} I_{(R,S,k)} M^k_{(r,s)}, \quad (4)$$

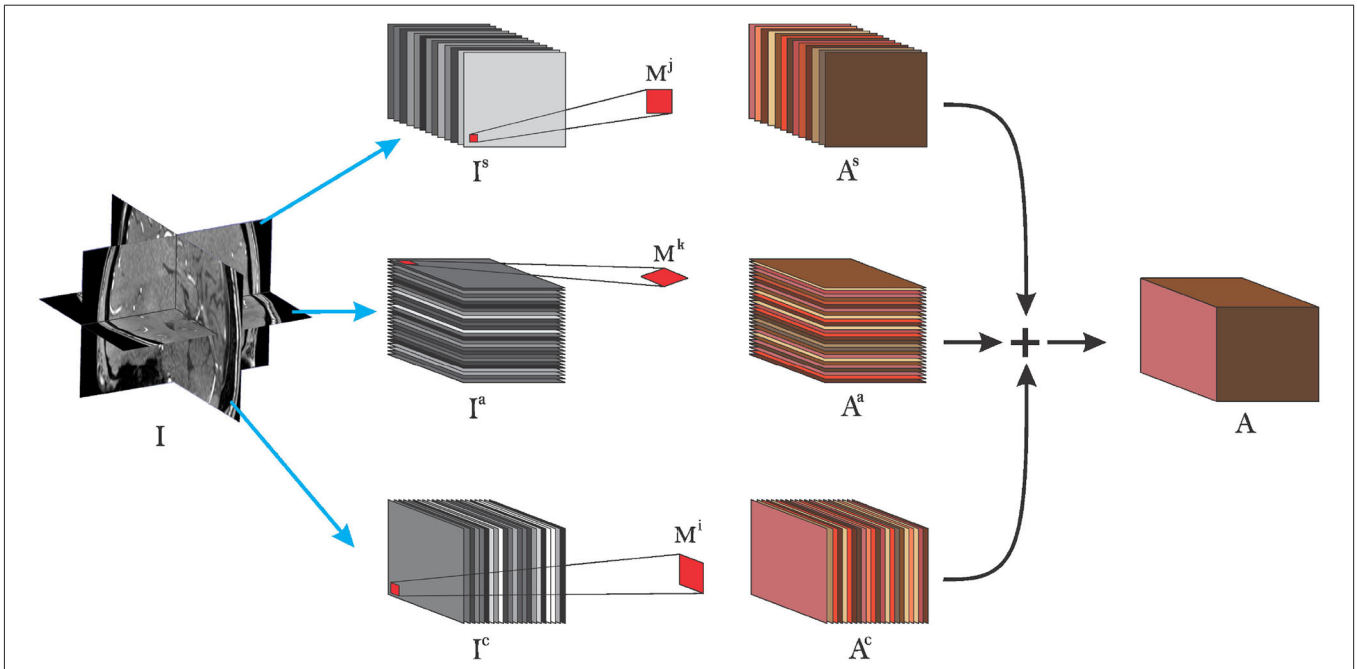


FIGURE 3 | Pictorial view of efficient implementation of cross-hair filters. Grayscaled stacks refer to input to the layer, red shaped squares refer to 2-D kernels used for each plane. Brown colored slices refer to extracted features after convolution operations and + symbol refers to matrix addition.

where M^i, M^j, M^k are 2-D convolutional (cross-hair) kernels used as an approximation to the 3-D kernel M in (1) along the ith, jth, and kth axes, respectively. R, S, and T are as defined in (1). Using cross-hair filters results in $(k_y k_z + k_x k_z + k_x k_y)$ multiplications and $(k_y k_z + k_x k_z + k_x k_y - 1)$ additions. If we let k_{m1}, k_{m2}, k_{m3} be the sizes of the kernel M such that $k_{m1} \geq k_{m2} \geq k_{m3}$, we can show that

$$k_y k_z + k_x k_z + k_x k_y \leq 3(k_{m1} k_{m2}) \leq k_x k_y k_z, \tag{5}$$

where strict inequality holds for all $k_{m3} > 3$. Equation (5) shows a better scaling in speed and also in memory since the filters sizes in (1) and (4) are affected by the same inequality. With the approximation in (4), and using the same example as above (volume of size $128 \times 128 \times 128$ and a $5 \times 5 \times 5$ kernel), we now need $<158 \times 10^6$ multiplications and 156×10^6 additions to compute the convolution leading to a reduction in computation by more than 100×10^6 multiplications and additions when compared to a classical 3-D convolution. Increasing the volume or kernel size, further increases the gap between the computational complexity of (1) and (4). Moreover, we will see later from our experiments that (4) still retains essential 3-D context information needed for the classification task.

2.1.1. Efficient Implementation

In Equation (4), we presented our 2-D crosshair filters. However, applying (4) independently for each voxel leads to a redundant use of memory. More precisely, voxels close to each other share some neighborhood information and making multiple copies of it is not memory efficient. To this end we present an efficient implementation below (depicted in **Figure 3**). Consider I as defined in Equation (1) and let us extract the sagital, coronal, and axial planes as I^s, I^c, and I^a, respectively. By application of Equations (1) and (4), we have a final implementation as follows:

$$\begin{aligned} I \diamond M &= A = \beta_c A^c + \beta_s A^s + \beta_a A^a, \\ A^c &= I^c ** M^i, \;\; A^s = I^s ** M^j, \;\; A^a = I^a ** M^k, \end{aligned} \tag{6}$$

where $**$ refers to a 2-D convolution along the first and second axes of the left matrix over all slices in the third axis. β_c, β_s, and β_a are weights to control the input of the planes toward the final sum, for example, in the case of different spatial resolutions of the planes (we use $\beta_c = \beta_s = \beta_a = 1$ in our experiments) and $\diamond$ refers to our crosshair filter operation. This implementation is efficient in the sense that it makes use of one volume at a time instead of copies of the volume in memory where voxels share the same neighborhood. In other words, we still have only one volume in memory but rather rotate the kernels to match the slices in the different orientations. This lowers the memory requirements during training and inference, allowing to train on more data with little memory.

2.1.2. 2.5-D Networks vs. 3-D Networks With Cross-Hair Filters

Its important to discuss the difference between existing 2.5-D networks and our proposed cross-hair filters. Given a 3-D task (e.g., vessel segmentation in 3-D volume), a 2.5-D based network handles the task by considering one 2-D slice at a time. More precisely, the network takes a 2-D slice (sometimes with few neighboring slices) as input and classifies all pixels in this

slice. This is repeated for each slice in the volume and the final results from the slices are fused again to form the 3-D result. On the architecture level, 2.5-D networks are 2-D networks with a preprocessing method for extracting 2-D slices and a postprocessing method for fusing 2-D results into a 3-D volume. We note that the predictions of 2.5-D networks are solely based on 2-D context information. Examples of 2.5-D networks are the implementation of UNet in Christ et al. (2016) used for liver and lesion segmentation tasks in CT volumetric dataset, and the network architecture of Sekuboyina et al. (2017) for annotation of lumbar vertebrae. Extensions of this approach may include a pre-processing stage where several 2-D planes are extracted and used as input channels to the 2-D network (Roth et al., 2014; Liu et al., 2017).

On the other hand, 3-D networks based on our proposed cross-hair filters take the whole 3-D volume as input, and at each layer in the network we apply the convolutional operator discussed in section 2.1. Therefore, our filters make use of 3-D context information at each convolutional layer and do not require specific preprocessing or post processing. Our proposed method differs from classical 3-D networks in the sense that it uses less parameters and memory since it does not use full 3-D convolutions. However, it is worth noting that our filters scale exactly the same as 2.5-D (i.e., in only two directions) with respect to changes in filter and volume sizes. More precisely, given a square or cubic filter of size k, we have k^2 parameters in a 2.5-D network and $3k^2$ in our cross-hair filter based network. Increasing the filter size by a factor of r will scale up as $k + r$ quadratically in both situations [i.e., $(k + r)^2$ for 2.5-D and $3(k + r)^2$ in cross-hair filter case] as compared to full 3-D networks where the parameter size scales as a cube of $k + r$.

In summary, unlike the existing 2.5-D ideas where 2-D planes are extracted at a pre-processing stage and used as input channels to a 2-D network architecture, our cross-hair filters are implemented on a layer level which help retain the 3-D information throughout the network making it a preferred option when detecting curvilinear objects in 3-D.

2.2. Extreme Class Balancing With Stable Weights

We now discuss the problem of "extreme" class imbalance and introduce a new cost function that is capable of dealing with this problem. Often in medical image analysis, the object of interest (e.g., vessel, tumor etc.) accounts for a minority of the total voxels of the image. The objects of interest in the datasets used in this work account for <2.5% of the voxels (the different datasets are described in section 3.1). A standard cross entropy loss function is given by

$$\mathcal{L}(\mathbf{W}) = -\frac{1}{N}\sum_{j=1}^{N} y_j \log P(y_j = 1|X; \mathbf{W}) + (1 - y_j) \log[1 - P(y_j = 1|X; \mathbf{W})], \quad (7)$$

$$\mathcal{L}(\mathbf{W}) = -\frac{1}{N}\left(\sum_{j\in Y_+} \log P(y_j = 1|X; \mathbf{W}) + \sum_{j\in Y_-} \log P(y_j = 0|X; \mathbf{W})\right),$$

where N is the total number of examples, P is the probability of obtaining the ground truth label given the data X and network weights $\mathbf{W}$, y_j is the label for the jth example, X is the feature set, W is the set of parameters of the network, Y_+ is the set of positive labels, and Y_- is the set of negative (background) labels. Using this cost function with extreme class imbalance between Y_- and Y_+ could cause the training process to be biased toward detecting background voxels at the expense of the object of interest. This normally results in predictions with high precision against low recall. To remedy this problem, Hwang and Liu (2015) proposed a biased sampling loss function for training multiscale convolutional neural networks for a contour detection task. This loss function introduced additional trade-off parameters and it samples twice more edge patches than non-edge ones for positive cost-sensitive finetuning, and vice versa, for negative cost-sensitive finetuning. Based on this, Xie and Tu (2015) proposed a class-balancing cross entropy loss function of the form

$$\mathcal{L}(\mathbf{W}) = -\beta \sum_{j\in Y_+} \log P(y_j = 1|X; \mathbf{W}) - (1 - \beta) \sum_{j\in Y_-} \log P(y_j = 0|X; \mathbf{W}), \quad (8)$$

where $\mathbf{W}$ denotes the standard set of parameters of the network, which are trained with backpropagation and β and $1 - \beta$ are the class weighting multipliers, which are calculated as $\beta = \frac{|Y_-|}{|Y|}$, $1 - \beta = \frac{|Y_+|}{|Y|}$. $P(.)$ is the probability from the final layer of the network, and Y_+ and Y_- are the set of positive and negative class labels, respectively.

2.2.1. Challenges From Numerical Instability and High False Positive Rate

The idea of giving more weight to the cost associated with the class with the lowest count from Equation (8), has been used in other recent works (Christ et al., 2016; Maninis et al., 2016; Nogues et al., 2016; Roth et al., 2016). However, our experiments (in section 3.4) show that the above loss function raises two main challenges.

First, there is the problem of numerical instability. The gradient computation is numerically unstable for very big training sets due to the high values taken by the loss. More precisely, there is a factor of $\frac{1}{N}$, that scales the final sum to the mean cost in the standard cross-entropy loss function in Equation (7). This factor ensures that the gradients are stable irrespective of the size of the training data N. However, in Equation (8), the weights β and $1 - \beta$ do not scale the cost to the mean value. For high number of data points $|Y|$ (which is usually the case of voxel-wise tasks), the sums explode leading

to numerical instability. For example, given a perfectly balanced data, we have $\beta = 1 - \beta = 0.5$, irrespective of the number of data points $|Y|$. Thus, increasing the size of the dataset (batch size) has no effect on the weights (β) but increases the number of elements in the summation, causing the computations to be unstable.

Second, there are challenges from high false positive rate. A high rate of false positives leading to high recall values is observed during training and at test time. This is caused by the fact that in most cases the object of interest accounts for <5% of the total voxels (about 2.5% in our case). Therefore, we have a situation where $1 - \beta < 0.05$, which implies that wrongly predicting 95 background voxels as foreground is less penalized in the loss than predicting 5 foreground voxels as background. This leads to high false positive rate and, hence, high recall values.

2.2.2. A New "Extreme" Class Balancing Function

To address the challenges discussed above, we introduce different weighting ratios and an additional factor to take care of the high false positive rate; and define:

$$
\begin{aligned}
\mathcal{L}(\mathbf{W}) &= \mathcal{L}_1(\mathbf{W}) + \mathcal{L}_2(\mathbf{W}) \qquad (9)\\
\mathcal{L}_1(\mathbf{W}) &= -\frac{1}{|Y_+|}\sum_{j\in Y_+}\log P(y_j = 1|X;\mathbf{W})\\
&\quad - \frac{1}{|Y_-|}\sum_{j\in Y_-}\log P(y_j = 0|X;\mathbf{W})\\
\mathcal{L}_2(\mathbf{W}) &= -\frac{\gamma_1}{|Y_+|}\sum_{j\in Y_{f+}}\log P(y_j = 0|X;\mathbf{W})\\
&\quad - \frac{\gamma_2}{|Y_-|}\sum_{j\in Y_{f-}}\log P(y_j = 1|X;\mathbf{W})\\
\gamma_1 &= 0.5 + \frac{1}{|Y_{f+}|}\sum_{j\in Y_{f+}}|P(y_j = 0|X;\mathbf{W}) - 0.5|\\
\gamma_2 &= 0.5 + \frac{1}{|Y_{f-}|}\sum_{j\in Y_{f-}}|P(y_j = 1|X;\mathbf{W}) - 0.5|
\end{aligned}
$$

where Y_{f+} and Y_{f-} are the set of false positive and false negative predictions respectively and |.| is the cardinality operator which measures the number of elements in the set. $\mathcal{L}_1$ is a more numerically stable version of Equation (8) since it computes the voxel-wise, cost which scales well with the size of the dataset or batch. But the ratio of β to $1 - \beta$ is maintained as desired. $\mathcal{L}_2$ [false prediction (FP) Rate Correction] is introduced to penalize the network for false predictions. However, we do not want to give false positive (Y_{f+}) and false negatives (Y_{f-}) the same weight as total predictions (Y_+,Y_-), since we will end up with a loss function without any class balancing because the weights will offset each other. Therefore, we introduce γ_1 and γ_2, which depend on the mean absolute distance of the wrong predicted probabilities from 0.5 (the value can be changed to suit the task). This allows us to penalize false predictions, which are very far from the central point (0.5). The false predictions (Y_{f+}, Y_{f-}) are obtained through a 0.5 probability threshold. Experimental results from application of FP rate correction can be found in section 3.3.2.

2.3. Synthetic Data for Transfer Learning

To generate synthetic data, we follow the method of Schneider et al. (2012) which implements a simulator of a vascular tree that follows a generative process inspired by the biology of angiogenesis. This approach, described in Schneider et al. (2012), has initially been developed to complement missing elements of a vascular tree, a common problem in μCT imaging of the vascular bed (Schneider et al., 2014). We now use this generator to simulate physiologically plausible vascular trees that we can use in training our CNN algorithms. The simulator considers the mutual interplay of arterial oxygen (O_2) supply and vascular endothelial growth factor (VEGF) secreted by ischemic cells to achieve physiologically plausible results. Each vessel segment is modeled as a rigid cylindrical tube with radius r and length l. It is represented by a single directed edge connecting two nodes. Semantically, this gives rise to four different types of nodes, namely root, leaf, bifurcation, and inter nodes. Each node is uniquely identified by the 3-D coordinate $\overrightarrow{P} = (x, y, z)^T$. Combining this with connectivity information, fully captures the geometry of the approximated vasculature. The tree generation model and the bifurcation configuration is shown in **Figure 4**. The radius of parent bifurcation branch r_p, and the radius of left (r_l) and right (r_r) daughter branches are related by a bifurcation law (also known as Murray's law) given by $r_p^\gamma = r_l^\gamma + r_r^\gamma$, where γ is the bifurcation exponent. Our simulator enforces the Murray's law during the tree generation process. Further constraints, $\cos(\phi_l) = \frac{r_p^4 + r_l^4 - r_r^4}{2r_p^2 r_l^2}$ and $\cos(\phi_r) = \frac{r_p^4 + r_r^4 - r_l^4}{2r_p^2 r_r^2}$ are placed on the bifurcation angles of the left (ϕ_l) and right (ϕ_r) vessel extension elements respectively. This corresponds corresponds to the optimal position of the branching point $\overrightarrow{P}_b$ with respect to a minimum volume principle, another constraint enforced in the simulator from Schneider et al. (2012, 2014).

2.3.1. Properties of the Simulated Data

The output of the generation process is a tree with information on the 3-D position $\overrightarrow{P}$ of the nodes, their type (root, bifircation, inter, leaf), and connectivity information, which includes the edge E_{ij} between two nodes N_i and N_j, and its radius R_{ij}. We reconstruct a 3-D volumetric data from this abstracted network description by modeling each vessel segment as a cylinder in 3-D space. We simulate different background and foreground intensity patterns with different signal-to-noise ratios. Detailed description of generated data is given in section 3.1.

3. EXPERIMENTS, RESULTS, AND DISCUSSION

3.1. Datasets

In this work, we use three different datasets to train and test the networks. In all three data sets, the test cases are kept apart from the training data and are used only for testing purposes. The

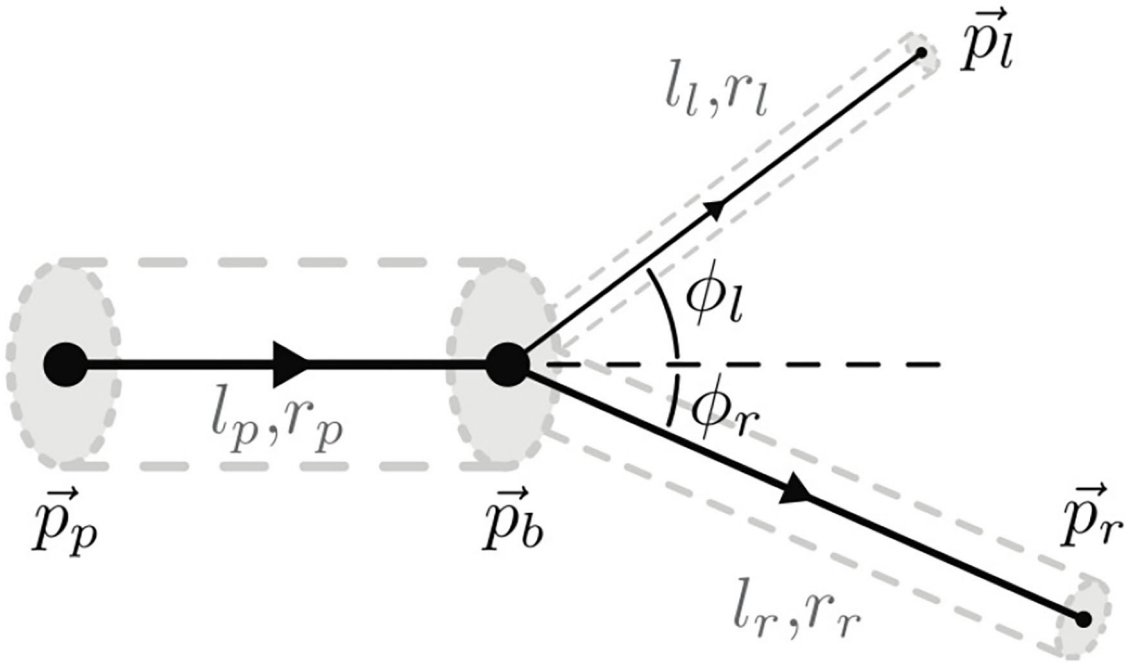


FIGURE 4 | A representation of the constrained bifurcation configuration, as presented in Schneider et al. (2012), where l_p, l_r, and l_l are the length of the parent, right daughter, and left daughter segments, respectively. p_r and p_l are the right and left daughter nodes, respectively.

datasets can be downloaded for public research from the paper's GitHub page (Tetteh, 2019a).

3.1.1. Synthetic Dataset

Training convolutional networks from scratch typically requires significant amounts of training data. However, assembling a properly labeled dataset of 3-D curvilinear structures, such as vessels and vessel features, takes a lot of human effort and time, which turns out to be the bottleneck for most medical applications. To overcome this problem, we generate synthetic data based on the method proposed in Schneider et al. (2012, 2014). A brief description of this process has already been presented in section 2.3. In the arterial tree generation experiment, the parameters in Table 1 of Schneider et al. (2012) are used. We use the default (underlined) values for all model parameters. We initialize the processes with different random seeds and scale the resulting vessel sizes in voxels to match the sizes of vessels in clinical datasets. Vessel intensities are randomly chosen in the interval $[128, 255]$ and non-vessel intensities are chosen from the interval $[0 - 100]$. Gaussian noise is then applied to the generated volume randomly changing the mean (i.e., in the range $[-5, 5]$) and the standard deviation (i.e., in the range $[-15, 30]$) for each volume. We generate 136 volumes of size $325 \times 304 \times 600$ with corresponding labels for vessel segmentation, centerlines, and bifurcation detection. Vessel, centerline and bifurcation labels occupy 2.1, 0.2, and 0.05% of total intensities, respectively, further highlighting the problem of class imbalance. Twenty volumes out of the 136 is used as a test set and the remaining volumes are used for pre-training our networks in the various tasks at hand. An example of the synthetic dataset can be found in **Figure 5C**. The synthetic dataset including both training and test volumes with ground truth labels for vessel, centerlines, and bifurcation are available at Tetteh (2019b) for download and public use.

3.1.2. Clinical Magnetic Resonance Angiograph (MRA) Dataset

To finetune and test our network architectures on real data, we obtain 40 volumes of clinical TOF MRA of the human brain, 20 of which are fully annotated, and the remaining 20 partially annotated using the method proposed by Forkert et al. (2013). Each volume has a size of $580 \times 640 \times 136$ and spacial resolution of $0.3125 \times 0.3125 \times 0.6mm$ on the coronal, sagittal, and axial axes, respectively. We select 15 out of the 20 fully annotated volumes for testing and use the remaining five as a validation set. We also correct the 20 partially annotated volumes by manually verifying some of the background and foreground voxels. This leads to three labels, which are true foreground (verified foreground), true background (verified background), and the third class, which represent the remaining voxels not verified. In our later experiments, we use the true foreground and background labels to finetune our networks. This approach helps in avoiding any uncertainty with respect to using the partially annotated data for finetuning of the network. Image intensity ranges were scaled with a quadratic function to enhance bright structures and normalized to a standard range after clipping high

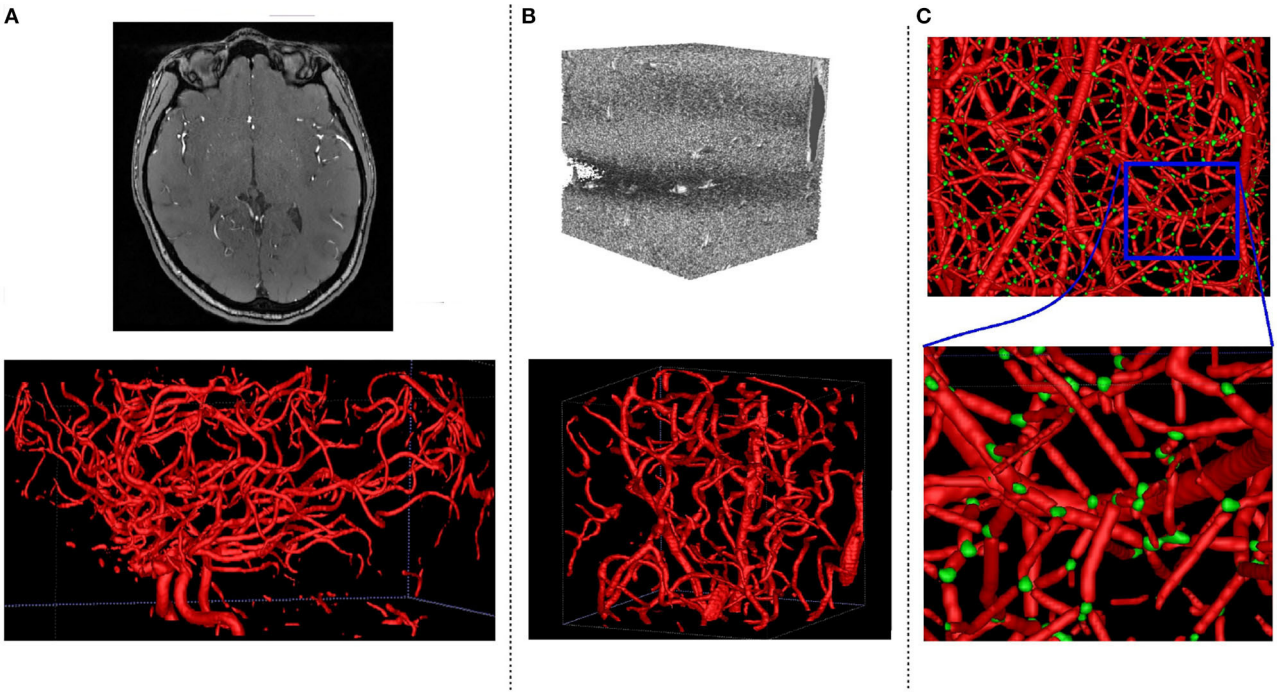


FIGURE 5 | Sample of datasets used in our experiments with the corresponding ground truth segmentations. **(A)** Clinical MRA, **(B)** μCTA, **(C)** Synthetic.

intensities. A sample volume of the TOF MRA dataset can be found in **Figure 5A**.

3.1.3. Micro Computed Tomogaphy Angiography (μCTA)

A 3-D volume of size $2,048 \times 2,048 \times 2,740$ and spacial resolution $0.7 \times 0.7 \times 0.7mm$ is obtained from synchrotron radiation X-ray tomographic microscopy of a rat brain. From this large volume, we extract a dataset of 20 non-overlaping volumes of size $256 \times 256 \times 256$, which were segmented using the method proposed by Schneider et al. (2015), and use them in our later experiments to finetune the networks. To create a test set, we manually annotate 52 slices in 4 other volumes different from the 20 volumes above (208 slices in total). As with the clinical MRA data, image intensity ranges for the μCTA were also scaled with a quadratic function to enhance bright structures and normalized to a standard range after clipping high intensities. Detailed description of the μCTA data can be found in Reichold et al. (2009), and a sample volume is presented in **Figure 5B**.

3.2. Network Architecture and Implementations

In this study we focus on the use of artificial neural networks for the tasks of vessel segmentation, centerline prediction, and bifurcation detection. Different variants of state-of-the-art Fully Convolutional Neural Networks have been presented for medical image segmentation (Christ et al., 2016; Maninis et al., 2016; Milletari et al., 2016; Nogues et al., 2016; Roth et al., 2016; Sekuboyina et al., 2017; Tetteh et al., 2017). Most of these architectures were based on the popular idea of convolutional-deconvolutional network which applies down-sampling at the earlier layers of the network and then reconstruct the volume at the later layers through up-sampling. This may be a bad choice given that the vascular tree tasks, especially centerline prediction and bifurcation detection, require fine details at the voxel level which can easily be lost through down-sampling. We therefore use a fully convolutional network (FCN) without down-sampling and up-sampling layers as a preferred architecture to test the performance of DeepVesselNet discussed in sections 2.1, 2.2, and 2.3. Nonetheless we also implement DeepVesselNet with popular convolutional-deconvolutional architectures to systematically study the effect of cross-hair kernel, as well as training behavior and generalization. Python implementation of our cross-hair filters and all other codes used in our experiments is available on GitHub (Tetteh, 2019a) for public use.

3.2.1. DeepVesselNet-FCN

We construct a Fully Convolutional Network FCN with four convolutional layers and a sigmoid classification layer. In this implementation, we do not use down-sampling and up-sampling layers and we carry out the convolutions in a way that the output image is of the same size as the input image by zero-padding. The removal of the down-sampling and up-sampling layers is motivated by the fact that the tasks (vessel segmentation, centerline prediction, and bifurcation detection) involve fine detailed voxel sized objects and down-sampling has an effect of averaging over voxels which causes these fine

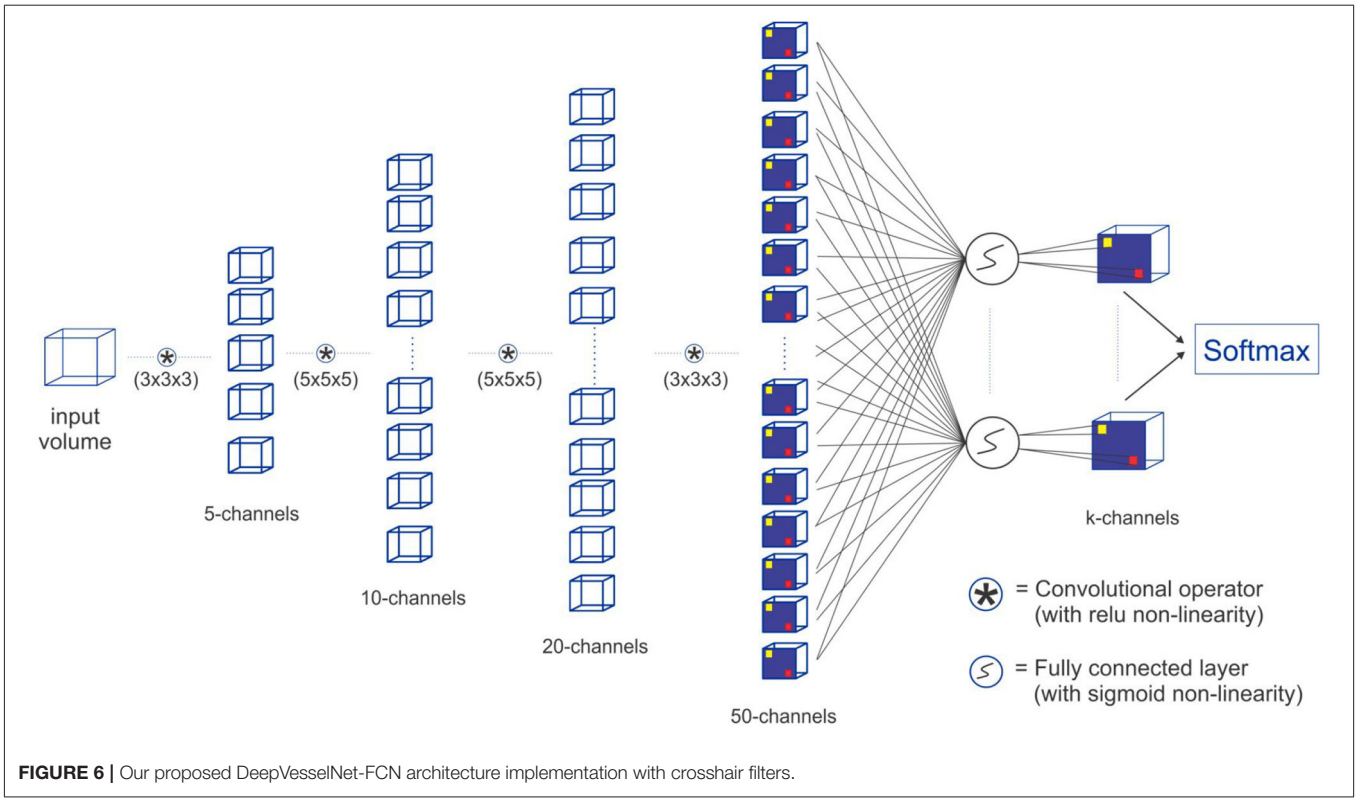


FIGURE 6 | Our proposed DeepVesselNet-FCN architecture implementation with crosshair filters.

details to be lost. The alternative max-intensity pooling can easily change the voxel position of the maximum intensity later in the up-sampling stage of the network. With DeepVesselNet-FCN implementation, we have a very simple 5-layer fully-convolutional network, which takes a volume of arbitrary size and outputs a segmentation map of the same size. For the network structure and a description of the parameters (see **Figure 6**).

3.2.2. DeepVesselNet-VNet and DeepVesselNet-Unet

To analyse the properties of our proposed cross-hair filters, we implement two alternative convolutional-deconvolutional architectures—VNet (Milletari et al., 2016) and 3D UNet (Çiçek et al., 2016)—and replace all 3-D convolutions with our proposed cross-hair filters discussed in section 2.1 to obtain DeepVesselNet-VNet and DeepVesselNet-UNet, respectively. By comparing the parameter size and execution time of DeepVesselNet-VNet and DeepVesselNet-UNet to the original VNet and 3D UNet implementations, we can evaluate the improvement in memory usage as well as the gain in speed that cross-hair filters offer. We also use these implementations to test whether gain in speed and memory consumption have a significant effect on prediction accuracy. Finally, DeepVesselNet-VNet and DeepVessel-UNet architectures include sub-sampling (down-sampling and up-sampling) layers. By comparing these two architecture with DeepVesselNet-FCN we can evaluate the relevance of sub-sampling when handling segmentation of fine structures like vessels and their centerlines and bifurcations.

3.2.3. Network Configuration, Initialization, and Training

We use the above described architecture to implement three binary networks for vessel segmentation, centerline prediction, and bifurcation detection. Network parameters are randomly initialized, according to the method proposed in Bengio and Glorot (2010), by sampling from a uniform distribution in the interval $(-\frac{1}{\sqrt{k_x k_y k_z}}, \frac{1}{\sqrt{k_x k_y k_z}})$ where $(k_x \times k_y \times k_z)$ is the size of the given kernel in a particular layer. For each volume in our training set, we extract non-overlapping boxes of size $(64 \times 64 \times 64)$ covering the whole volume and then feed them through the network for the finetuning of parameters. The box extraction is only done at training time to enable fast training and efficient use of computation memory, this is however not needed after our convolutional kernels are trained since full volumes can be used at test time. We train the network using a stochastic gradient descent optimizer without regularization. During pre-training, we use a learning rate of 0.01 and decay of 0.99, which is applied after every 200 iterations for all network architectures. For finetuning, we use a learning rate of 0.001 and a decay of 0.99 applied after every 200 iterations. We implement our algorithm using the THEANO (Theano Development Team, 2016) Python framework and train on a machine with 64GB of RAM and Nvidia TITAN X 12GB GPU.

3.3. Evaluating the DeepVesselNet Components

Prior to evaluating the performance of DeepVesselNet, we conducted a series of experiments to test the components of

DeepVesselNet which includes fast cross-hair filters, the FP rate correction, and pre-training on synthetic data. Results of this ablation analysis are offered here.

3.3.1. Fast Cross-Hair Filters

To investigate the usefulness of cross-hair filters in DeepVesselNet, we experiment with full 3-D versions of DeepVesselNet and evaluate the effect on performance based on three main criteria—memory footprint based on number of parameters, computational speed based on execution time, and prediction accuracy based on Dice score. **Table 1** shows the number of parameters in the various architectures and the execution times in the three datasets. Comparing DeepVesselNet-VNet and DeepVesselNet-UNet with their 3-D versions (VNet and UNet), we find more than 27% (16.56 vs. 22.89 m and 4.45 vs. 7.41 m, respectively) reduction in memory footprint. Also, the execution time in **Table 1** shows that cross-hair filters improve the computational speed of DeepVesselNet-VNet and DeepVesselNet-UNet by more than 23% over VNet and UNet respectively in both synthetic and clinical MRA datasets. DeepVesselNet-FCN uses very low (only 0.05 m) number of parameters as compared to the other architectures due to the absence of sub-sampling layers. Scores in **Table 1** are obtained using kernels of size $3 \times 3 \times 3$ and $5 \times 5 \times 5$, and the benefits of using sparse cross-hair filter will be even more profound with larger kernel sizes and volume sizes. Evaluation of cross-hair filters in terms of prediction accuracy is discussed in section 3.4.

3.3.2. Extreme Class Balancing ($\mathcal{L}_1 + \mathcal{L}_2$)

To test the effect of FP rate correction loss function discussed in section 2.2, we train the DeepVesselNet-FCN architecture on a sub-sample of four clinical MRA volumes from scratch, with and without FP rate correction described in Equation (9). We train for 5,000 iterations and record the ratio of precision to recall every 5 iterations using a threshold of 0.5 on the probability maps. A plot of the precision-recall ratio during training without FP rate correction ($\mathcal{L}_1$ Only) and with FP rate correction ($\mathcal{L}_1 + \mathcal{L}_2$) is

TABLE 1 | Number of convolutional parameters in the networks used in our experiments.

Architecture	Params (millions)	Ex. time Synthetic (s)	Ex. time TOF MRA (s)	Ex. time μCTA (s)
DeepVesselNet-FCN	0.05	13	13	4
DeepVesselNet-VNet	16.56	17	20	7
DeepVesselNet-UNet	4.45	13	14	4
VNet	22.89	23	26	11
UNet	7.41	17	19	6

For the purpose of comparison, the number of parameters stated here refers to only the convolutional layers in the various architectures. Ex. Time refers to the average time in seconds required to process one volume in the sythetic and MRA TOF datasets.

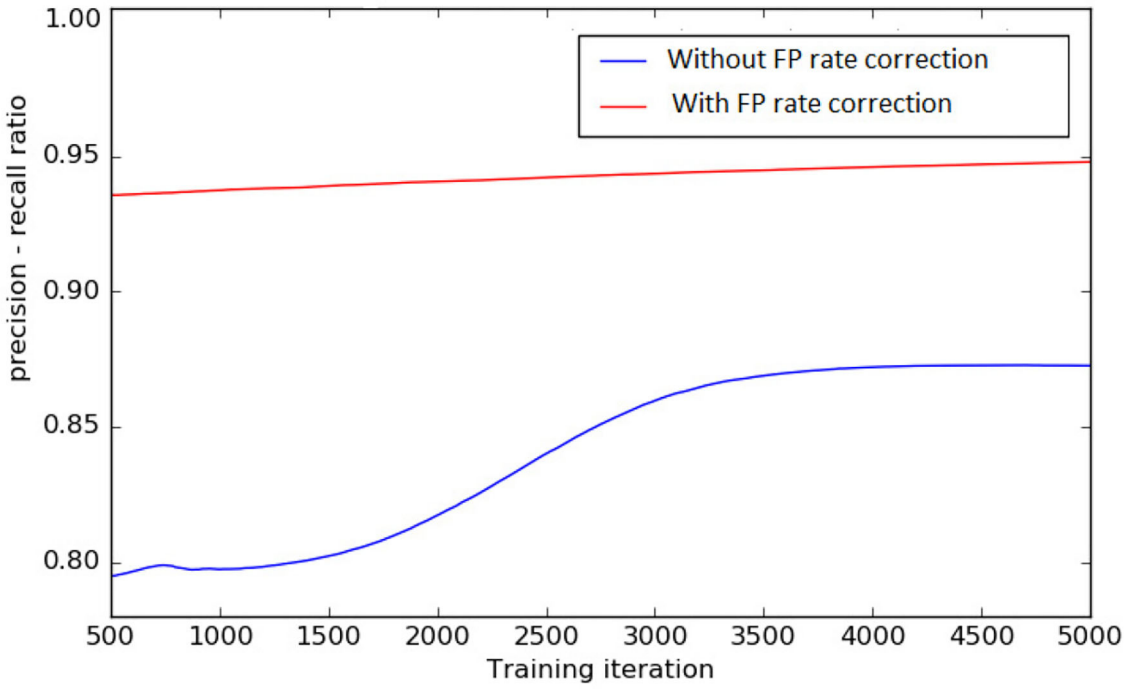


FIGURE 7 | Precision—recall ratio during training, with FP rate correction and without FP rate correction in the loss function, on four selected clinical MRA volumes. A balanced precision-recall ratio (i.e., close to 1) implies that we obtain the FP rate correction we propose in the work and the training process is not bias toward the background or the foreground.

presented in **Figure 7**. The results of this experiments suggest that training with both factors in the loss function, as proposed in section 2.2, keeps a better balance between precision and recall (i.e., a ratio closer to 1.0) than without the second factor. A balanced precision-recall ratio implies that the training process is not bias toward the background or the foreground. This helps prevent over-segmentation, which normally occurs as a result of the introduction of the class balancing.

3.3.3. Transfer Learning From Synthetic Data

We assess the usefulness of transfer learning with synthetic data by comparing the training convergence speed, and various other scores that we obtain when we pretrain DeepVesselNet-FCN on synthetic data and finetune on the clinical MRA dataset, compared to training DeepVesselNet-FCN from scratch on the clinical MRA. For this experiment, we only consider the vessel segmentation task, as no annotated clinical data is available for centerline and bifurcation tasks. Results of this experiment are reported in **Table 2**. We achieve a Dice score of 86.39% for training from scratch without pre-training on synthetic data and 86.68% when pretraining on synthetic data. This shows that training from scratch or pre-training on synthetic data does not make much difference regarding the accuracy of the results. However, training from scratch requires about 600 iterations more than pre-training on synthetic data for the network to converge (i.e., 50% more longer).

TABLE 2 | Results from pretraining DeepVesselNet-FCN on synthetic data and finetuning with the training set from the clinical MRA vs. training DeepVesselNet-FCN from scratch on clinical MRA.

Method	Precision	Recall	Dice	Iterations
With pretraining	86.44	86.93	86.68	1200
Without pretraining	85.87	86.92	86.39	1800

Iterations refers to training iterations required for the network to converge. Although the result in Dice score are not very different, it is clear that the pre-training on synthetic data leads to an earlier convergence of the network.

3.4. Evaluating DeepVesselNet Performance

In this subsection, we retain the best training strategy from the described experiments in section 3.3 and assess the performance of our proposed network architecture with other available methods mainly on the vessel segmentation task. As a further validation of our methodology we handle centerline prediction and bifurcation detection using the proposed architectures. Given a good vessel segmentation, centerline prediction, and bifurcation detection tasks is classically handled by applying vessel skeletonization as a post processing step and a search of the resulting graph. Our aim in applying our architecture to handle these tasks is not to show superiority over the existing vessel skeletonization methods but it is to serve as a further verification of the effects of our described methodology and to offer a complementary way of obtaining centerlines and bifurcations, for example, to increase the robustness of the processing pipeline when fusing results of complementary approaches. The details of these experiments, results and discussion are given below.

TABLE 3 | Results for vessel segmentation.

Dataset	Method	Prec.	Rec.	Dice
Synthetic	DeepVesselNet-FCN	**99.84**	**99.87**	**99.86**
	DeepVesselNet-VNet	99.54	99.59	99.56
	DeepVesselNet-UNet	99.48	99.42	99.45
	VNet	99.48	99.50	99.49
	UNet	99.57	99.52	99.55
	Schneider et al.	99.47	99.56	99.52
TOF MRA	DeepVesselNet-FCN (finetuned)	**86.44**	**86.93**	**86.68**
	DeepVesselNet-FCN (pretrained)	82.76	80.25	81.48
	DeepVesselNet-VNet (finetuned)	85.00	83.51	84.25
	DeepVesselNet-VNet (pretrained)	83.32	77.12	80.10
	DeepVesselNet-UNet (finetuned)	83.56	85.18	84.36
	DeepVesselNet-UNet (pretrained)	83.48	79.27	81.32
	VNet (finetuned)	84.34	85.62	84.97
	VNet (pretrained)	82.41	75.82	78.98
	UNet (finetuned)	84.02	85.35	84.68
	UNet (pretrained)	83.16	80.23	81.67
	Schneider et al.	84.81	82.15	83.46
	Forkert et al.	84.99	73.00	78.57
μCTA	DeepVesselNet-FCN	**96.72**	95.82	**96.27**
	DeepVesselNet-VNet	95.83	**96.18**	96.01
	DeepVesselNet-UNet	95.85	96.06	95.95
	VNet	95.25	95.84	95.55
	UNet	95.27	95.71	95.49
	Schneider et al.	95.15	91.51	93.30

TOF MRA are evaluated within the brain region only.
Pretrained results refers to the scores we obtained on the test set after pretraining, and finetuned results are scores after finetuning with annotated data available for TOF-MRA. Best performing method in each metric are show in bold.

3.4.1. Vessel Segmentation

We pretrain DeepVesselNet-(FCN, VNet, UNet) architectures on synthetic volumes for vessel segmentation and evaluate its performance on TOF MRA volumes through a transfer learning approach. We later finetune the networks with additional clinical MRA data, repeating the evaluation. **Table 3** reports results of these tests, together with performances of competing methods. We obtain a Dice score of 81.48% for DeepVesselNet-FCN, 81.32% for DeepVessel-UNet, and 80.10% for DeepVesselNet-VNet on TOF MRA test dataset with the transfer learning step, and 86.68% (DeepVesselNet-FCN), 84.36% (DeepVesselNet-UNet) as well as 84.25% (DeepVesselNet-VNet) after finetuning. This results (also box plots in **Figure 8**) suggest that, with a Cox-Wilcoxon significance test $p < 0.001$, DeepVesselNet-FCN which does not use sub-sampling outperforms the versions of networks that use sub-sampling layers (VNet and UNet). **Table 3** also reports results from the methods of Schneider et al. (2015) and Forkert et al. (2013) all of which are outperformed by DeepVesselNet-FCN in terms of Dice score. Comparing DeepVesselNet-VNet and VNet (84.25 vs. 84.97% with a p-value

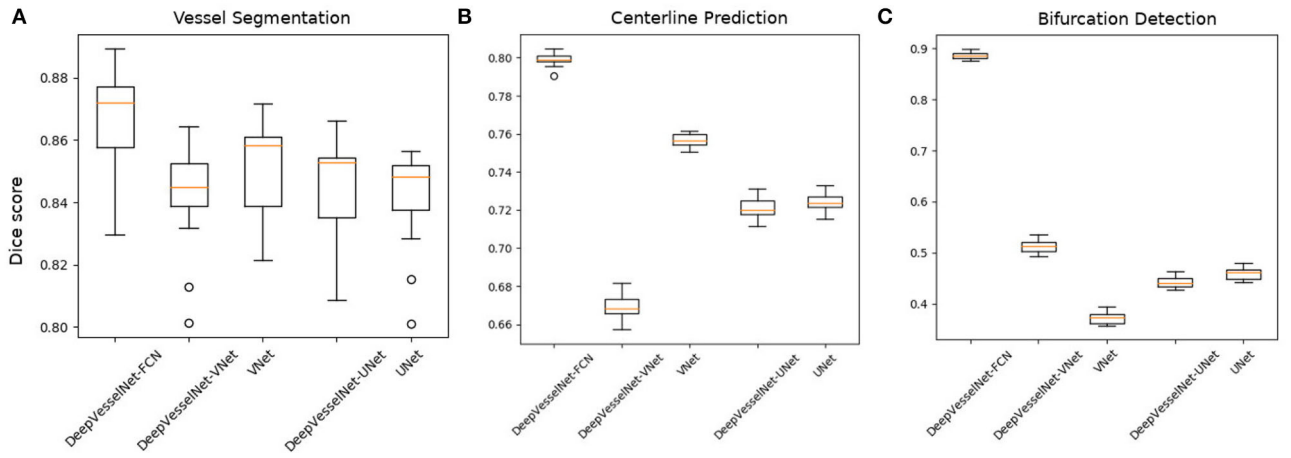


FIGURE 8 | Box plots of Dice scores from vessel segmentation **(A)**, centerline prediction **(B)**, and bifurcation detection **(C)** tasks over our test set in the clinical MRA dataset across the deep learning architectures. It is evident that DeepVesselNet-VNet and DeepVesselNet-UNet obtain comparable results as VNet and UNet, respectively. However, DeepVesselNet-FCN achieves a significantly higher results in all three tasks as confirmed by a $p < 0.001$.

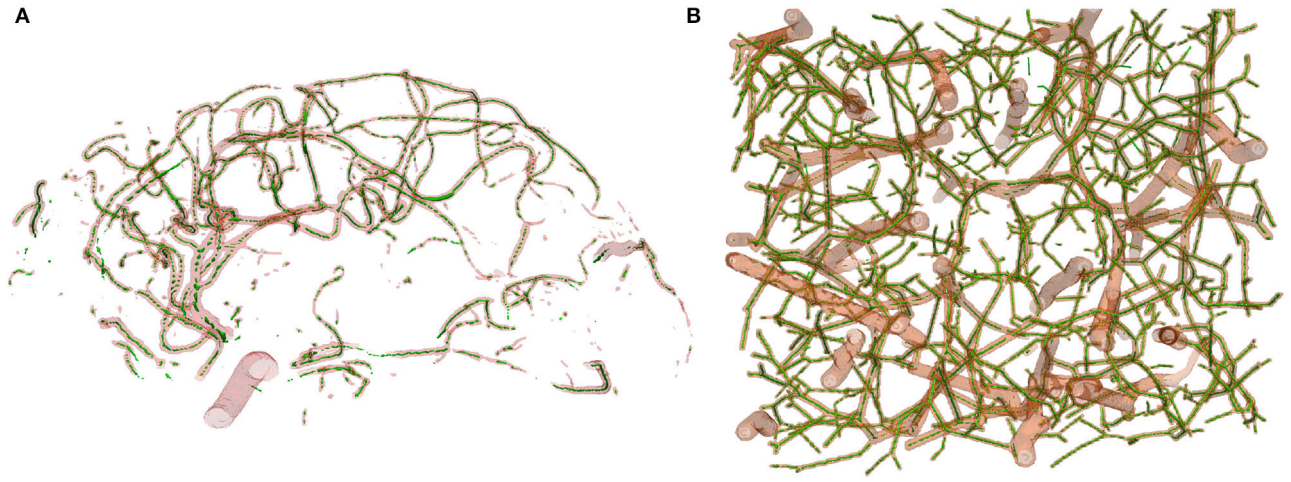


FIGURE 9 | Centerline prediction on MRA TOF **(A)** and Synthetic **(B)** test datasets using DeepVesselNet-FCN (centerline in green). There are more detections in smaller vessels than in larger vessels which can be explained by the network seeing more smaller vessels than bigger vessels during training. Again, centerline detection in MRA TOF covers all the vessels with missing points and can be improved by finetuning on annotated MRA data or by a post-processing strategy to fill in the missing points.

of 0.04) as well as DeepVesselNet-Unet and UNet (84.36 vs. 84.68 with a p-value of 0.07) on the MRA data, we find an advantage of up to 1% for the latter in terms of Dice scores. However, DeepVesselNet-VNet and DeepVesselNet-Unet have the advantage of being memory and computationally efficient as seen in **Table 1**. These results show that cross-hair filters can be used in DeepVesselNet at a little to no cost in terms of vessel segmentation accuracy.

3.4.2. Centerline Prediction

For centerline prediction, we train DeepVesselNet on the synthetic dataset, test it on synthetic dataset and present visualizations on synthetic and clinical MRA datasets (see **Figure 9**). The networks use the probabilistic segmentation masks from the vessel segmentation step as an input. Qualitative results are presented in **Figure 9** together with quantitative scores in **Table 4**. DeepVesselNet-VNet performs slightly worse than VNet in terms of the Dice score (66.96 vs. 74.82% with a p-value of 0.0001). Similar trend can be seen when we compare Dice scores of DeepVesselNet-UNet and UNet (72.10 vs. 72.41% with a p-value of 0.0001). We obtain a Dice score of 79.92% for DeepVesselNet-FCN, which outperforms UNet and VNet and their corresponding DeepVesselNet variants with a significance test $p < 0.0001$. Here we note that the morphological operations based method of Schneider et al., which represents a state of the art method for centerline prediction, is able to obtain

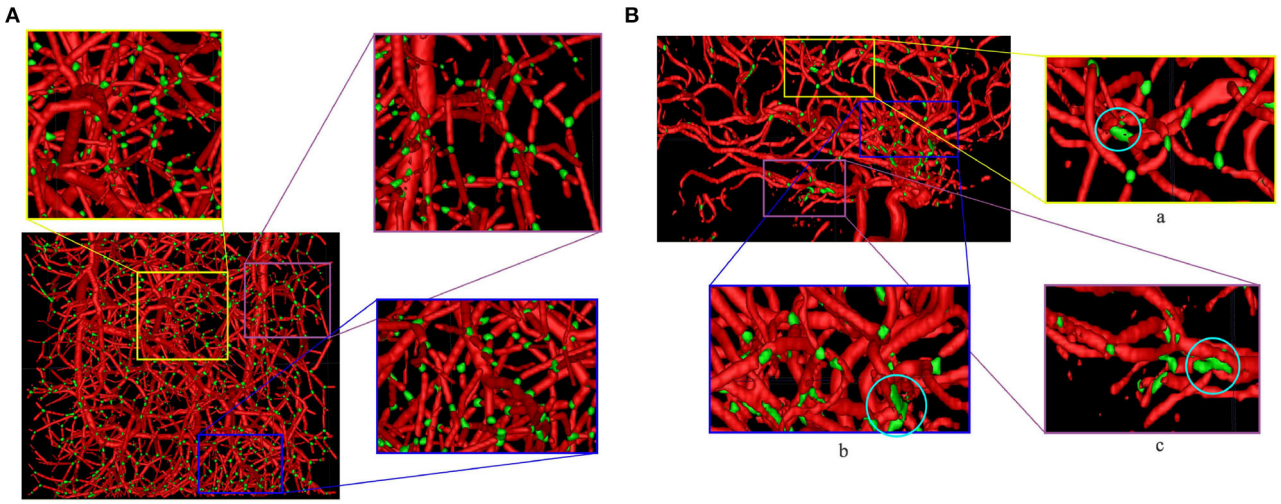


FIGURE 10 | Bifurcation detection on synthetic **(A)** and MRA TOF **(B)** test datasets using DeepVesselNet-FCN (bifurcations in green). Similar to centerline prediction, bifurcation detections in smaller vessels are better than in bigger vessels which might be due to the network seeing more examples in smaller vessels than in bigger vessels during training. At regions where a lot of vessels intersect, the network predicts it as a big bifurcation, this can be seen in the circled regions in zoomed images (a–c).

a higher recall than DeepVesselNet-FCN method (86.03 vs. 82.35%). This means that it detects more of the centerline points than DeepVesselNet-FCN. However, it suffers from lower precision (48.07 vs. 77.63%) due to higher false positive rate which causes the overall performance to fall (61.68 vs. 79.92% Dice score) as compared to DeepVesselNet-FCN. From the box plots in **Figure 8** it is very evident DeepVesselNet-FCN significantly outperforms all other architectures suggesting that the performance of the other architectures suffers from the use of sub-sampling layers.

3.4.3. Bifurcation Detection

For a quantitative evaluation of DeepVesselNet in bifurcation detection, we use synthetically generated data, and adopt a two-input-channels strategy. We use the vessel segmentations as one input channel and the centerline predictions as a second input channel relying on the same training and test splits as in the previous experiments. In our predictions we aim at localizing a cubic region of size ($5 \times 5 \times 5$) around the bifurcation points, which are contained within the vessel segmentation. We evaluate the results based on a hit-or-miss criterion: a bifurcation point in the ground truth is counted as hit if a region of a cube of size ($5 \times 5 \times 5$) centered on this point overlaps with the prediction, and counted as a miss otherwise; a hit is considered as true positive (TP) and a miss is considered as false negative (FN); a positive label in the prediction is counted as false positive (FP) if a cube of size ($5 \times 5 \times 5$) centered on this point contains no bifurcation point in the ground truth. Qualitative results on synthetic and clinical MRA TOF are shown in **Figure 10**, respectively. Results for Schneider et al. are obtained by first extracting the vessel tree and searching the graph for nodes. Then all nodes with two or more splits are treated as bifurcations—this being one of the standard methods for bifurcation extraction. In **Figure 8**, we present the box plots of Dice score distributions obtained by the different architectures over our test set. Results from **Table 5** and **Figure 8** show that DeepVesselNet-FCN performs better than the other architectures in 5 out of 6 metrics. In our experiments, it became evident that VNet tends to over-fit, possibly due to its high number of parameters. This may explain why results for VNet are worse than all other methods, also suggesting that in cases where little training data is available, the DeepVesselNet-FCN architecture may be the preferable due to low number of parameters and the absence of sub-sampling layers.

TABLE 4 | Results for centerline prediction tasks.

Method	Prec.	Rec.	Dice
DeepVesselNet-FCN	**77.63**	82.35	**79.92**
DeepVesselNet-VNet	65.15	68.87	66.96
DeepVesselNet-UNet	71.28	72.95	72.10
VNet	76.41	73.30	74.82
UNet	71.25	73.61	72.41
Schneider et al.	48.07	**86.03**	61.68

Results suggest that architectures with sub-sampling layers suffer fall in performance due to loss of fine details which is crucial in centerline prediction.
Best performing methods in each category in bold.

4. SUMMARY AND CONCLUSIONS

We present DeepVesselNet, an architecture tailored to the challenges of extracting vessel networks and features using deep learning. Our experiments in sections 3.3 and 3.4 show that the cross-hair filters, which is one of the components of DeepVesselNet, performs comparably well as 3-D filters and, at

TABLE 5 | Results from bifurcation detection experiments.

Method	Prec.	Rec.	Det. %	Mean err	Err std
DeepVesselNet-FCN	**78.80**	**92.97**	**86.87**	0.2090	**0.6671**
DeepVesselNet-VNet	46.80	56.70	84.21	1.6533	0.9645
DeepVesselNet-UNet	29.47	88.41	85.89	0.6227	0.9380
VNet	25.50	68.71	70.29	1.2434	1.3857
UNet	32.57	77.81	71.78	1.2966	1.4000
Schneider et al.	77.18	85.08	84.30	**0.1529**	0.7074

Precision and recall are measured on the basis of the 5×5×5 blocks around the bifurcation points. Mean error and its corresponding standard deviation are measured in voxels away from the bifurcation points (not the 5 × 5 × 5 blocks).
Best performing method in each metric are show in bold.

the same time, improves significantly both speed and memory usage, easing an upscaling to larger data sets. Another component of DeepVesselNet, the introduction of new weights and the FP rate correction discussed in section 2.2, helps in maintaining a good balance between precision and recall during training. This turns out to be crucial for preventing over and under-segmentation problems, which are common problems in vessel segmentation. We also show from our results in section 3.4 that using sub-sampling layers in a network architecture in tasks which includes voxel-sized objects can lead to a fall in performance. Finally, we successfully demonstrated in sections 3.3 and 3.4 that transfer learning of DeepVesselNet through pre-training on synthetically generated data improves segmentation and detection results, especially in situations where obtaining manually annotated data is a challenge.

As future work, we will generalize DeepVesselNet to multiclass vessel tree task, handling vessel segmentation, centerline prediction, and bifurcation detection simultaneously, rather than in three subsequent binary tasks. We also expect that network architectures tailored to our three hierarchically nested classes will improve the performance of the DeepVesselNet. For example by using a multi-level activation approach proposed in Piraud et al. (2019) or through a single, but hierarchical approach starting from a base network for vessel segmentation, additional layers for centerline prediction, and a final set of layers for bifurcation detection.

The current implementation of cross-hair filters, network architectures and cost function are available on GitHub (Tetteh, 2019a). Datasets can also be downloaded from the wiki page of the same GitHub page.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found at: https://github.com/giesekow/deepvesselnet/wiki/Datasets.

AUTHOR CONTRIBUTIONS

GT, VE, and MP contributed to experiments and writing of the manuscript. NF, MS, and JK contributed toward data generation and preparation. BW, CZ, and BM contributed toward experiment and manuscript verification, and supervisory role. All authors contributed to the article and approved the submitted version.

ACKNOWLEDGMENTS

This manuscript has been released as a pre-print at: https://arxiv.org/abs/1803.09340 (Tetteh et al., 2018).

REFERENCES

Bengio, Y., and Glorot, X. (2010). "Understanding the difficulty of training deep feedforward neuralnetworks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)* (Sardinia), *Vol. 9*.

Chaichana, T., Sun, Z., Barrett-Baxendale, M., and Nagar, A. (2017). "Automatic location of blood vessel bifurcations in digital eye fundus images," in *Proceedings of Sixth International Conference on Soft Computing for Problem Solving: SocProS 2016, Vol. 2* (Singapore: Springer Singapore), 332–342. doi: 10.1007/978-981-10-3325-4_33

Chen, D., and Cohen, L. D. (2015). "Piecewise geodesics for vessel centerline extraction and boundary delineation with application to retina segmentation," in *Scale Space and Variational Methods in Computer Vision: 5th Int. Conf., SSVM 2015* (Lége-Cap Ferret: Springer International Publishing), 270–281. doi: 10.1007/978-3-319-18461-6_22

Christ, P. F., Elshaer, M. E. A., Ettlinger, F., Tatavarty, S., Bickel, M., Bilic, P., et al. (2016). "Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3d conditional random fields," in *Proc. MICCAI 2016* (Cham: Springer International Publishing), 415–423. doi: 10.1007/978-3-319-46723-8_48

Chung, A. C. S., and Noble, J. A. (1999). "Statistical 3d vessel segmentation using a Rician distribution," in *Proc. MICCAI 1999*, eds C. Taylor and A. Colchester (Berlin; Heidelberg: Springer), 82–89. doi: 10.1007/10704282_9

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). 3D u-net: Learning dense volumetric segmentation from sparse annotation. *CoRR* abs/1606.06650. doi: 10.1007/978-3-319-46723-8_49

Ciresan, D. C., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2012). "Deep neural networks segment neuronal membranes. Electron microscopy images," in *NIPS* (Nevada, NV), 2852–2860.

Dalca, A., Danagoulian, G., Kikinis, R., Schmidt, E., and Golland, P. (2011). "Segmentation of nerve bundles and ganglia in spine MRI using particle filters," in *Proc. MICCAI 2011* (Berlin; Heidelberg: Springer), 537–545. doi: 10.1007/978-3-642-23626-6_66

Florin, C., Paragios, N., and Williams, J. (2006). "Globally optimal active contours, sequential Monte Carlo and on-line learning for vessel segmentation," in *Computer Vision-ECCV 2006: 9th European Conf. on Computer Vision, Graz, Austria* (Berlin; Heidelberg: Springer), 476–489.

Forkert, N. D., Schmidt-Richberg, A., Fiehler, J., Illies, T., Möller, D., Handels, H., et al. (2011). Fuzzy-based vascular structure enhancement in time-of-flight MRA images for improved segmentation. *Methods Inform. Med.* 50, 74–83. doi: 10.3414/ME10-02-0003

Forkert, N. D., Schmidt-Richberg, A., Fiehler, J., Illies, T., Möller, D., Säring, D., et al. (2013). 3d cerebrovascular segmentation combining fuzzy vessel enhancement and level-sets with anisotropic energy weights. *Magn. Reson. Imaging* 31, 262–271. doi: 10.1016/j.mri.2012.07.008

Frangi, A. F., Niessen, W. J., Vincken, K. L., and Viergever, M. A. (1998). "Multiscale vessel enhancement filtering," in *Proc. MICCAI 1998*, eds W. M.

Wells, A. Colchester, and S. Delp (Berlin; Heidelberg: Springer), 130–137. doi: 10.1007/BFb0056195

Grzymala-Busse, J. W., Goodwin, L. K., Grzymala-Busse, W. J., and Zheng, X. (2004). "An approach to imbalanced data sets based on changing rule strength," in *Rough-Neural Computing: Techniques for Computing With Words*, eds S. K. Pal, L. Polkowski, and A. Skowron (Berlin; Heidelberg: Springer), 543–553. doi: 10.1007/978-3-642-18859-6_21

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017). Learning from class-imbalanced data: review of methods and applications. *Expert Syst. Appl.* 73, 220–239. doi: 10.1016/j.eswa.2016.12.035

Hwang, J.-J., and Liu, T.-L. (2015). "Pixel-wise deep learning for contour detection," in *ICLR* (San Diego, CA).

Kirbas, C., and Quek, F. (2004). A review of vessel extraction techniques and algorithms. *ACM Comput. Surv.* 36, 81–121. doi: 10.1145/1031120.1031121

Koziński, M., Mosinska, A., Salzmann, M., and Fua, P. (2018). "Learning to segment 3d linear structures using only 2d annotations," in *Medical Image Computing and Computer Assisted Intervention-MICCAI 2018*, eds A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger (Cham: Springer International Publishing), 283–291. doi: 10.1007/978-3-030-00934-2_32

Law, M. W. K., and Chung, A. C. S. (2008). "Three dimensional curvilinear structure detection using optimally oriented flux," in *Computer Vision-ECCV 2008*, eds D. Forsyth, P. Torr, and A. Zisserman (Berlin; Heidelberg: Springer), 368–382. doi: 10.1007/978-3-540-88693-8_27

Lesage, D., Angelini, E., Bloch, I., and Funka-Lea, G. (2009). A review of 3d vessel lumen segmentation techniques: models, features and extraction schemes. *Med. Image Anal.* 13, 819–845. doi: 10.1016/j.media.2009.07.011

Liao, W., Rohr, K., and Wörz, S. (2013). "Globally optimal curvature-regularized fast marching for vessel segmentation," in *Proc. MICCAI 2013* (Berlin; Heidelberg: Springer), 550–557. doi: 10.1007/978-3-642-40811-3_69

Liu, S., Zhang, D., Song, Y., Peng, H., and Cai, W. (2017). "Triple-crossing 2.5d convolutional neural network for detecting neuronal arbours in 3d microscopic images," in *Mach. Learn. in Med. Imaging* (Quebec, QC: Springer Int. Publishing), 185–193. doi: 10.1007/978-3-319-67389-9_22

Macedo, M. M. G., Mekkaoui, C., and Jackowski, M. P. (2010). "Vessel centerline tracking in CTA and MRA images using hough transform," in *Progress in Pattern Recognition, Image Anal., Computer Vision, and Applications: 15th Iberoamerican Congress on Pattern Recognition, CIARP 2010* (Berlin; Heidelberg: Springer), 295–302.

Maddah, M., Afzali-khusha, A., and Soltanian, H. (2003). Snake modeling and distance transform approach to vascular center line extraction and quantification. *Computer. Med. Imag. Graph.* 27, 503–512. doi: 10.1016/S0895-6111(03)00040-5

Maninis, K.-K., Pont-Tuset, J., Arbeláez, P., and Van Gool, L. (2016). "Deep retinal image understanding," in *Proc. MICCAI 2016, Part II* (Athens: Springer International Publishing), 140–148. doi: 10.1007/978-3-319-46723-8_17

Martínez-Pérez, M. E., Hughes, A. D., Stanton, A. V., Thom, S. A., Bharath, A. A., and Parker, K. H. (1999). "Retinal blood vessel segmentation by means of scale-space analysis and region growing," in *Proc. MICCAI 1999* (Berlin; Heidelberg: Springer), 90–97. doi: 10.1007/10704282_10

Milletari, F., Navab, N., and Ahmadi, S. (2016). "V-net: Fully convolutional neural networks for volumetric med. image segmentation," in *Fourth Int. Conf. on 3D Vision (3DV)* (Stanford), 565–571. doi: 10.1109/3DV.2016.79

Moreno, R., Wang, C., and Smedby, Ö. (2013). "Vessel wall segmentation using implicit models and total curvature penalizers," in *Image Anal.: 18th Scandinavian Conf., Proc.* (Berlin; Heidelberg: Springer), 299–308. doi: 10.1007/978-3-642-38886-6_29

Nain, D., Yezzi, A., and Turk, G. (2004). "Vessel segmentation using a shape driven flow," in *Proc. MICCAI 2004* (Berlin; Heidelberg: Springer), 51–59. doi: 10.1007/978-3-540-30135-6_7

Nogues, I., Lu, L., Wang, X., Roth, H., Bertasius, G., Lay, N., et al. (2016). "Automatic lymph node cluster segmentation using holistically-nested neural networks and structured optimization in CT images," in *Proc. MICCAI 2016, Part II* (Athens: Springer International Publishing), 388–397. doi: 10.1007/978-3-319-46723-8_45

Phellan, R., and Forkert, N. D. (2017). Comparison of vessel enhancement algorithms applied to time-of-flight MRA images for cerebrovascular segmentation. *Med. Phys.* 44, 5901–5915. doi: 10.1002/mp.12560

Phellan, R., Peixinho, A., Falc ao, A., and Forkert, N. D. (2017). "Vascular segmentation in TOF MRA images of the brain using a deep convolutional neural network," in *Intravascular Imaging and Computer Assisted Stenting, and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis* (Quebec: Springer International Publishing), 39–46. doi: 10.1007/978-3-319-67534-3_5

Piraud, M., Sekuboyina, A., and Menze, B. H. (2019). "Multi-level activation for segmentation of hierarchically-nested classes," in *Computer Vision-ECCV 2018 Workshops*, eds L. Leal-Taixé and S. Roth (Cham: Springer International Publishing), 345–353. doi: 10.1007/978-3-030-11024-6_24

Reichold, J., Stampanoni, M., Keller, A. L., Buck, A., Jenny, P., and Weber, B. (2009). Vascular graph model to simulate the cerebral blood flow in realistic vascular networks. *J. Cereb. Blood Flow Metab.* 29, 1429–1443. doi: 10.1038/jcbfm.2009.58

Rempfler, M., Schneider, M., Ielacqua, G. D., Xiao, X., Stock, S. R., Klohs, J., et al. (2015). Reconstructing cerebrovascular networks under local physiological constraints by integer programming. *Med. Image Anal.* 86–94. doi: 10.1016/j.media.2015.03.008

Rigamonti, R., Sironi, A., Lepetit, V., and Fua, P. (2013). "Learning separable filters," in *The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Portland, OR). doi: 10.1109/CVPR.2013.355

Roth, H. R., Lu, L., Farag, A., Sohn, A., and Summers, R. M. (2016). "Spatial aggregation of holistically-nested networks for automated pancreas segmentation," in *Proc. MICCAI 2016, Part II* (Athens: Springer International Publishing), 451–459. doi: 10.1007/978-3-319-46723-8_52

Roth, H. R., Lu, L., Seff, A., Cherry, K. M., Hoffman, J., Wang, S., et al. (2014). "A new 2.5d representation for lymph node detection using random sets of deep convolutional neural network observations," in *Proc. MICCAI 2014* (Boston, MA: Springer International Publishing), 520–527. doi: 10.1007/978-3-319-10404-1_65

Santamaría-Pang, A., Colbert, C. M., Saggau, P., and Kakadiaris, I. A. (2007). "Automatic centerline extraction of irregular tubular structures using probability volumes from multiphoton imaging," in *Proc MICCAI 2007, Part II* (Berlin; Heidelberg: Springer), 486–494.

Schneider, M., Hirsch, S., Weber, B., Székely, G., and Menze, B. (2014). "TGIF: topological gap in-fill for vascular networks-a generative physiological modeling approach," in *International Conference on Medical Image Computing and Computer-Assisted Intervention, Vol. 17*, 89–96. doi: 10.1007/978-3-319-10470-6_12

Schneider, M., Hirsch, S., Weber, B., Székely, G., and Menze, B. H. (2015). Joint 3-d vessel segmentation and centerline extraction using oblique hough forests with steerable filters. *Med. Image Anal.* 19, 220–249. doi: 10.1016/j.media.2014.09.007

Schneider, M., Reichold, J., Weber, B., Székely, G., and Hirsch, S. (2012). Tissue metabolism driven arterial tree generation. *Med. Image Anal.* 1397–1414. doi: 10.1016/j.media.2012.04.009

Sekuboyina, A., Valentinitsch, A., Kirschke, J., and Menze, B. H. (2017). A localisation-segmentation approach for multi-label annotation of lumbar vertebrae using deep nets. *arXiv arXiv*: 1703.04347.

Shagufta, B., Khan, S. A., Hassan, A., and Rashid, A. (2014). "Blood vessel segmentation and centerline extraction based on multilayered thresholding in CT images," in *Proc. of the 2nd Int. Conf. on Intelligent Systems and Image Processing* (Kitakyushu), 428–432.

Szczerba, D., and Székely, G. (2005). "Simulating vascular systems in arbitrary anatomies," in *Proc. MICCAI 2005* (Berlin; Heidelberg: Springer), 641–648. doi: 10.1007/11566489_79

Tetteh, G. (2019a). *Implementation of the Deepvesselnet Deep Learning Network*. Available online at: https://github.com/giesekow/deepvesselnet

Tetteh, G. (2019b). *Synthetic Dataset Used for Training of Deepvesselnet*. Available online at: https://github.com/giesekow/deepvesselnet/wiki/Datasets

Tetteh, G., Efremov, V., Forkert, N. D., Schneider, M., Kirschke, J., Weber, B., et al. (2018). Deepvesselnet: vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. *arXiv*. CoRR, abs/1803.09340.

Tetteh, G., Rempfler, M., Zimmer, C., and Menze, B. H. (2017). "Deep-fext: deep feature extraction for vessel segmentation and centerline prediction," in *Mach.*

Learn. in Med. Imaging (Quebec, QC: Springer International Publishing), 344–352. doi: 10.1007/978-3-319-67389-9_40

Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints: abs/1605.02688*.

Todorov, M. I., Paetzold, J. C., Schoppe, O., Tetteh, G., Shit, S., Efremov, V., et al. (2020). Machine learning analysis of whole mouse brain vasculature. *Nat. Methods* 17, 442–449. doi: 10.1038/s41592-020-0792-1

Wang, S., Peplinski, B., Lu, L., Zhang, W., Liu, J., Wei, Z., et al. (2013). "Sequential Monte Carlo tracking for marginal artery segmentation on CT angiography by multiple cue fusion," in *Proc. MICCAI 2013* (Berlin; Heidelberg: Springer), 518–525. doi: 10.1007/978-3-642-40763-5_64

Wörz, S., Godinez, W. J., and Rohr, K. (2009). "Probabilistic tracking and model-based segmentation of 3d tubular structures," in *Bildverarbeitung für die Medizin 2009: Algorithmen – Systeme – Anwendungen Proc. des Workshops* (Berlin; Heidelberg: Springer), 41–45. doi: 10.1007/978-3-540-93860-6_9

Xie, S., and Tu, Z. (2015). "Holistically-nested edge detection," in *Proceedings of the IEEE International Conference on Computer Vision* (Santiago), 1395–1403. doi: 10.1109/ICCV.2015.164

Young, S., Pekar, V., and Weese, J. (2001). "Vessel segmentation for visualization of MRA with blood pool contrast agent," in *Proc. MICCAI 2001* (Berlin; Heidelberg: Springer), 491–498. doi: 10.1007/3-540-45468-3_59

Zheng, Y., Liu, D., Georgescu, B., Nguyen, H., and Comaniciu, D. (2015). "3D deep learning for efficient and robust landmark detection in volumetric data," in *Proceedings of MICCAI 2015*, eds N. Navab, J. Hornegger, W. M. Wells, and A. Frangi (Munich: Springer International Publishing), 565–572. doi: 10.1007/978-3-319-24553-9_69

Zheng, Y., Shen, J., Tek, H., and Funka-Lea, G. (2012). "Model-driven centerline extraction for severely occluded major coronary arteries," in *Mach. Learn. in Medical Imaging: Third Int. Workshop, MLMI 2012, Held in Conjunction With MICCAI: Nice, France, Revised Selected Papers* (Berlin; Heidelberg: Springer), 10–18. doi: 10.1007/978-3-642-35428-1_2

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

5

A DEEP LEARNING APPROACH TO PREDICT COLLATERAL FLOW IN STROKE PATIENTS USING RADIOMIC FEATURES FROM PERFUSION IMAGES

Authors: **Giles Tetteh**, Fernando Navarro, Raphael Meier, Johannes Kaesmacher, Johannes C. Paetzold, Jan S. Kirschke, Claus Zimmer, Roland Wiest and Bjoern H. Menze.

Abstract: Collateral circulation results from specialized anastomotic channels which are capable of providing oxygenated blood to regions with compromised blood flow caused by arterial obstruction. The quality of collateral circulation has been established as a key factor in determining the likelihood of a favorable clinical outcome and goes a long way to determining the choice of a stroke care model. Though many imaging and grading methods exist for quantifying collateral blood flow, the actual grading is mostly done through manual inspection. This approach is associated with a number of challenges. First, it is time-consuming. Second, there is a high tendency for bias and inconsistency in the final grade assigned to a patient depending on the experience level of the clinician. We present a multi-stage deep learning approach to predict collateral flow grading in stroke patients based on radiomic features extracted from MR perfusion data. First, we formulate a region of interest detection task as a reinforcement learning problem and train a deep learning network to automatically detect the occluded region within the 3D MR perfusion volumes. Second, we extract radiomic features from the obtained region of interest through local image descriptors and denoising auto-encoders. Finally, we apply a convolutional neural network and other machine learning classifiers to the extracted radiomic features to automatically predict the collateral flow grading of the given patient volume as one of three severity classes - no flow (0), moderate flow (1), and good flow (2). Results from our experiments show an overall accuracy of 72% in the three-class prediction task. With an inter-observer agreement of 16% and a maximum intra-observer agreement of 74% in a similar experiment, our automated deep

learning approach demonstrates a performance comparable to expert grading, is faster than visual inspection, and eliminates the problem of grading bias.

Date of publication: 21 February 2023
Journal: Frontiers in Neurology

Contribution of thesis author: project conception and coordination, experimental design, algorithm development and implementation, data analysis, leading author of manuscript.

frontiers | Frontiers in Neurology

TYPE Original Research
PUBLISHED 21 February 2023
DOI 10.3389/fneur.2023.1039693

OPEN ACCESS

EDITED BY
Jean-Claude Baron,
University of Cambridge, United Kingdom

REVIEWED BY
Anna Podlasek,
University of Dundee, United Kingdom
Arvind Bambhroliya,
University of Texas Health Science Center at Houston, United States
Jennifer Soun,
University of California, Irvine, United States

*CORRESPONDENCE
Giles Tetteh
✉ giles.tetteh@tum.de

†PRESENT ADDRESS
Johannes C. Paetzold,
Department of Computing, Imperial College London, London, United Kingdom

SPECIALTY SECTION
This article was submitted to Stroke, a section of the journal Frontiers in Neurology

RECEIVED 08 September 2022
ACCEPTED 02 February 2023
PUBLISHED 21 February 2023

CITATION
Tetteh G, Navarro F, Meier R, Kaesmacher J, Paetzold JC, Kirschke JS, Zimmer C, Wiest R and Menze BH (2023) A deep learning approach to predict collateral flow in stroke patients using radiomic features from perfusion images. *Front. Neurol.* 14:1039693. doi: 10.3389/fneur.2023.1039693

A deep learning approach to predict collateral flow in stroke patients using radiomic features from perfusion images

Giles Tetteh[1,2]*, Fernando Navarro[1], Raphael Meier[3], Johannes Kaesmacher[3], Johannes C. Paetzold[1†], Jan S. Kirschke[2], Claus Zimmer[2], Roland Wiest[3] and Bjoern H. Menze[1,4]

[1]Department of Computer Science, Technische Universität München, München, Germany, [2]Neuroradiology, Klinikum Rechts der Isar, Technische Universität München, München, Germany, [3]Institute for Diagnostic and Interventional Neuroradiology, Inselspital University Hospital, Bern, Switzerland, [4]Department of Quantitative Biomedicine, University of Zurich, Zurich, Switzerland

Collateral circulation results from specialized anastomotic channels which are capable of providing oxygenated blood to regions with compromised blood flow caused by arterial obstruction. The quality of collateral circulation has been established as a key factor in determining the likelihood of a favorable clinical outcome and goes a long way to determining the choice of a stroke care model. Though many imaging and grading methods exist for quantifying collateral blood flow, the actual grading is mostly done through manual inspection. This approach is associated with a number of challenges. First, it is time-consuming. Second, there is a high tendency for bias and inconsistency in the final grade assigned to a patient depending on the experience level of the clinician. We present a multi-stage deep learning approach to predict collateral flow grading in stroke patients based on radiomic features extracted from MR perfusion data. First, we formulate a region of interest detection task as a reinforcement learning problem and train a deep learning network to automatically detect the occluded region within the 3D MR perfusion volumes. Second, we extract radiomic features from the obtained region of interest through local image descriptors and denoising auto-encoders. Finally, we apply a convolutional neural network and other machine learning classifiers to the extracted radiomic features to automatically predict the collateral flow grading of the given patient volume as one of three severity classes - no flow (0), moderate flow (1), and good flow (2). Results from our experiments show an overall accuracy of 72% in the three-class prediction task. With an inter-observer agreement of 16% and a maximum intra-observer agreement of 74% in a similar experiment, our automated deep learning approach demonstrates a performance comparable to expert grading, is faster than visual inspection, and eliminates the problem of grading bias.

KEYWORDS

collateral flow, radiomics, perfusion, reinforcement learning, image descriptors, angiography, auto-encoder, deep learning

1. Introduction

Collateral circulation results from specialized anastomotic channels which are present in most tissues and capable of providing nutrient perfusion to regions with compromised blood flow due to ischemic injuries caused by ischemic stroke, coronary atherosclerosis, peripheral artery disease, and similar conditions or diseases (1). Collateral circulation

helps to sustain blood flow in the ischaemic areas in acute, subacute, or chronic phases after an ischaemic stroke or transient ischaemic attack (2). The quality of collateral circulation has been convincingly established as a key factor in determining the likelihood of successful reperfusion and favorable clinical outcome (3). It is also seen as one of the major determinants of infarct growth in the early time windows which is likely to have an impact on the chosen stroke care model that is the decision to transport or treat eligible patients immediately.

A high number of imaging methods exist to assess the structure of the cerebral collateral circulation and several grading criteria have been proposed to quantify the characteristics of collateral blood flow. However, this grading is mostly done through visual inspection of the acquired images which introduces two main challenges.

First, there are *biases and inconsistencies in the current grading approaches*: There is a high tendency of introducing bias in the final grade assigned to a patient depending on the experience level of the clinician. There are inconsistencies also in the grade assigned by a particular clinician at different times for the same patient. These inconsistencies are quantified at 16% interobserver agreement and a maximum intraobserver agreement of 74% respectively in a similar study by Ben Hassen et al. (4).

Second, *grading is time-consuming and tedious*: Aside the problem of bias prediction, it also takes the clinician several minutes to go through the patient images to first select the correct image sequence, detect the region of collateral flow and then to be able to assign a grading a period of time which otherwise could have been invested in the treatment of the patient.

In this work, we analyze several machine learning and deep learning strategies that aim toward automating the process of collateral circulation grading. We present a set of solutions focusing on two main aspects of the task at hand.

First, *the region of interest needs to be identified*. We automate the extraction of the region of interest (ROI) from the patient images using deep reinforcement learning (RL). This is necessary for achieving a fully automated system that will require no human interaction and save the clinician the time spent on performing this task.

Finally, *the region of interest needs to be processed and classified*. We consider various feature extraction schemes and classifiers suitable for the task described above. This helps to extract useful image features, both learned and hand-crafted, which are relevant to the classification task. We predict digitally subtracted angiography (DSA) based collateral flow grading from MR perfusion images in this task. This saves the time required in choosing the right DSA sequence from the multiple DSA sequences acquired and helps achieve a fully automated system.

1.1. Prior work and open challenges

1.1.1. Imaging criteria for cerebral collateral circulation

Imaging methods for assessing cerebral collateral flow can be grouped under two main classification schemes, invasive vs. non-invasive and structural vs. functional imaging. Structural imaging methods provide information about the underlying structure of the cerebral collateral circulation network. Some of the commonly used structural imaging modalities are traditional single-phase computed tomography angiography (CTA), time-of-flight magnetic resonance angiography (TOF-MRA), and digitally subtracted angiography (DSA), among others. Other imaging modalities have been used in clinical practice and relevant research areas in accessing the structure of the cerebral collateral circulation are discussed in Liu et al. (2), McVerry et al. (5), Martinon et al. (6). DSA is the gold standard for assessing the collateral flow, however, due to the associated high cost and invasive nature, other non-invasive methods like CTA and MRA are commonly used (2).

Functional imaging methods are used to assess the function of the underlying cerebral collateral circulation. Single-photon emission CT (SPECT), MR perfusion, and positron emission tomography (PET) are examples of imaging methods that provide functional information about the cerebral collateral flow. MR perfusion imaging is often followed by a post-processing step to extract parametric information. Very common parametric information includes the *time-to-peak* (T_{max}) time taken for the blood flow to reach its peak (max) at a given region in the brain, *relative blood flow* (rBF) volume of blood flowing through a given brain tissue per unit of time, and *relative blood volume* (rBV) volume of blood in a given brain tissue relative to an internal control (e.g. normal white matter or an arterial input function). Functional imaging is sometimes combined with structural imaging either in a single scanning procedure or separate procedures and can serve as complementing information in the decision making process. Here, structural imaging is oftentimes used to map the anatomy and probe tissue microstructure.

MRI perfusion and diffusion have evolved as key biomarkers in determining collateralization of stroke patients, and a patient stratification based on these markers has been proposed repeatedly (7). At the same time, a qualitative CTA and DSA based grading are the most common approaches for evaluating collateralization (8–10).

1.1.2. Cerebral collateral flow grading

Cerebral collateral circulation plays an important role in stabilizing cerebral blood flow when the normal blood circulation system is compromised in cases of acute, subacute, or chronic ischaemic stroke. The quality of the cerebral collateral circulation system is one of the factors that determine the speed of infarct growth and the outcome of stroke treatment and reperfusion therapies. A poor collateral flow is associated with worse outcomes and faster growth of infarcts while a good collateral flow is associated with good outcomes and slower growth of infarcts in acute stroke treatment (11). Due to the important role played by cerebral collateral blood flow, various grading scales and their association with risk factors and treatment outcomes have been discussed extensively in literature.

Several grading systems have been proposed for assessing the quality of the collateral circulation network. Among these grading systems, the DSA based system proposed by the American Society of Interventional and Therapeutic Neuroradiology/Society of Interventional Radiology (ASITN/SIR) is the most widely

accepted scheme. This grading system describes the collateral flow as one of five levels of flow which are; absence of collaterals (0), slow collaterals (1), rapid collaterals (2), partial collaterals (3), and complete collaterals (4) to the periphery of the ischaemic site (2, 12). In most studies that use the ASITN/SIR scheme, the grading scale is merged into three levels—grades 0–1 (poor), 2 (moderate) and 3–4 (good collateral) flow. CTA based systems also have several grading schemes ranging from two (good, bad) to five (absent, diminished >50%, <50%, equal, more) labels (12).

The relationship between pretreatment collateral grade and vascular recanalization has been assessed for patients who received endovascular therapy for acute cerebral ischemia from two distinct study populations by Bang et al. (13). The study showed that 14.1, 25.2, and 41.5% of patients with poor, good, and excellent pretreatment collaterals respectively achieved complete revascularization. Another study by Bang et al. (14) on the relationship between MRI diffusion and perfusion lesion indices, angiographic collateral grade, and infarct growth showed that the greatest infarct growth occurred in patients with both non-recanalization and poor collaterals. Mansour (15) assessed collateral pathways in acute ischemic stroke using a new grading scale (Mansour Scale) and correlated the findings with different risk factors, clinical outcomes, and recanalization rates with endovascular management. More research (13–17) has been conducted into the relationship between the cerebral collateral circulation, its grading, and the clinical outcome of the choice of treatment of acute ischemic stroke, and they all confirm a positive association between collateral flow and the success of the outcome.

Due to the crucial role played by collateral circulation, it is a common practice in most clinical procedures to determine the quality of a patient's collateral as first-hand information toward the choice of the treatment or care model. This grading is done manually by inspecting patient scans which is time-consuming and also introduces some level of bias in the final grade assigned to a patient. Ben Hassen et al. (4) evaluated the inter-and intraobserver agreement in angiographic leptomeningeal collateral flow assessment on the ASITN/SIR scale and found an overall interobserver agreement $\kappa = 0.16 \pm 6.5 \times 10^{-3}$ among 19 observers with grades 0 and 1 being associated with the best results of $\kappa = 0.52 \pm 0.001$ and $\kappa = 0.43 \pm 0.004$ respectively. By merging the scales into two classes, poor collaterals (grade 0, 1, or 2), versus good collaterals (grade 3 or 4), the interobserver agreement increased to $\kappa = 0.27 \pm 0.014$. The same study recorded maximum intraobserver agreements of $\kappa = 0.74 \pm 0.1$ and $\kappa = 0.79 \pm 0.11$ for the ASITN/SIR and dichotomized scales respectively. McHugh (18) presented a study on interrater reliability and the kappa statistic as a measure of agreement and recommended a moderate interobserver agreement of $0.60 \leq \kappa \leq 0.79$ as a minimum requirement for medical data and study. These results are evidence of the need to automate the collateral grading process to achieve speed and consistency in the assigned grading.

Methods for automating the grading of collateral flow have not yet been properly explored in literature. Kersten-Oertel et al. (19) presented an automated technique to compute a collateral circulation score based on differences seen in mean intensities between left and right cerebral hemispheres in 4D angiography images and found a good correlation between the computed score and radiologist score ($r^2 = 0.71$) and good separation between good and intermediate/poor groups. Grunwald et al. (20) used a machine learning approach to categorize the degree of collateral flow in 98 patients who were eligible for mechanical thrombectomy and generated an e-CTA collateral score (CTA-CS) for each patient. The experiments showed that the e-CTA generated improved the intraclass correlation coefficient between three experienced neuroradiologists from 0.58 (0.46–0.67) to 0.77 (0.66–0.85, $p = 0.003$).

1.1.3. Reinforcement learning for medical imaging

Defining the region of interest (ROI) is often the first step in most image-based radiomics pipelines. This is the case because full patient scans often include artifacts and other information which are irrelevant and can affect the final outcome of the study. Therefore, most pipelines propose a manual localization of a ROI as a preprocessing step. However, it is crucial to define the ROI in an automated and reproducible fashion in other to achieve a fully automated pipeline. We propose a reinforcement learning approach for the localization of the region of interest due to increased speed and lower training data requirements compared to other supervised learning approaches.

Reinforcement learning (RL) has become one of the most active research areas in machine learning and involves the training of a machine learning agent to make a sequence of reward-based decisions toward the achievement of a goal through interaction with the environment. The idea of RL has been long applied in the field of robotics for robot vision and navigation (21–23) before the topic became very popular in the image processing society. RL has been used in the general field of computer vision mainly for object detection (24–26), image segmentation (27, 28), and image enhancement (29–31). However, in medical imaging RL is still in the research phase with very high potential. Netto et al. (32) presented an overview of medical imaging applications applying reinforcement learning with a detailed illustration of a use case involving lung nodules classification which showed promising results. Sahba et al. (27) implemented an RL based thresholding for segmenting prostate in ultrasound images with results that showed high potential for applying RL in medical image segmentation. Alansary et al. (33) evaluated reinforcement learning agents for anatomical landmark detection by comparing fixed and multi-scale search strategies with hierarchical action steps in a coarse-to-fine manner and achieved a performance better than state-of-the-art supervised learning methods.

1.2. Main contributions

In this study, we employ parametric information (T_{max}, rBF, rBV) from MR perfusion images of patients with acute ischaemic stroke and predict the three-level DSA based grading of these patients based on this functional information. We hypothesize that the rich information on blood flow visible from MRI perfusion can be used to predict collateral flow in a similar manner to DSA. Moreover, we argue that this approach, using 3D information, may even offer a more reliable biomarker than the interpretation of DSA images. As collateralization patterns are often unstable and

may undergo significant changes in the course of minutes, a second estimate of the activation of collateral flow using MRI—in addition to the subsequent DSA—will offer better diagnostic information.

We explore machine learning and deep learning methods in collateral flow grading. We apply deep reinforcement learning, a variant of RL which combines the power of deep learning and reinforcement learning, to detect a rigid-sized cube around the occluded region in an acute ischemic stroke patient scan as an initial step toward the prediction of cerebral collateral flow grading. This step is necessary to automate the detection of the occluded region which improves the accuracy of the prediction. Reducing the time spent on this task and ensuring that the proposed methodology is fully automated.

We provide experiments on different feature extraction strategies including denoising autoencoder (DAE), histogram of oriented gradient (HOG), and local binary pattern (LBP). The extracted features are further utilized in a random forest (RF), K-nearest neighbor (KNN), support vector machine (SVM), and convolutional neural network (CNN) classifiers for the prediction of the collateral flow grading. We provide detailed experimental setup and results which will serve as a guide for further research in this direction.

2. Methodology

In this section, we will discuss the details of the steps we employed in predicting the collateral flow grading from MR perfusion data. Figure 1 shows an overview of the main steps involved in the classification process. The first step is the detection of the region of interest (ROI) using reinforcement learning. This step helps to narrow down the classification task to only the area which has been occluded from normal blood flow. The second step deals with extracting features from the ROI. Finally, we feed the extracted features to a set of classifiers to obtain the collateral flow grading for the given patient data.

2.1. Deep reinforcement learning for region of interest detection

The idea of reinforcement learning includes an artificial agent which is trained to interact with an environment through a sequence of reward-based decisions toward a specific goal. At every time step t, the agent takes into account its current state s and performs an action a in a set of actions A and receives a reward r which is a measure of how good or bad the action a is toward the achievement of the set goal. The aim of the agent, which is to find an optimal policy (set of states, actions, and rewards) that maximizes the future reward, can be formulated as a Markov Decision Process. Since Markov Decision Process involves a large number of possible decision points which are normally not fully observable, RL approximates the optimal decision function by iteratively sampling from the set of policies through a process known as Q-learning.

2.1.1. Q-learning

At time point t and state s, let $\pi = a_i^{t+T}_{i=t}$ be a policy that is a sequence of actions needed by the agent to move from the current state s to the target. Let Q_t be a future discounted reward function such that

$$Q_t(s,\pi) = \sum_{i=t}^{t+T} \gamma^{i-t} r_{\pi i}, \tag{1}$$

where $r_{\pi i}$ is the reward associated with the action a_i of policy π at time $t = i, \gamma \in [0,1]$ is the future reward discounting factor, and T is the number of steps needed to reach the target by the chosen policy π. At any time step t the optimal policy π^* is the policy that maximizes the expected value of Q_t. This can be represented by an action-value function $Q_t^*(s)$ defined by

$$\pi^* = Q_t^*(s) = \max_{\pi} \mathbb{E}[Q_t(s,\pi)] \tag{2}$$

The optimal value function $Q_t^*(s)$ obeys the Bellman equation, stating that if the optimal value $Q_{t+1}^*(s)$ of the next state is known for all possible policies π, then the optimal behavior is to select the policy π^* that maximizes the expected value of $r_{\pi t} + Q_{t+1}(s,\pi)$ [which follows from setting $i = t$ in Equation (1)]. The action-value function can therefore be estimated recursively as

$$Q_t^*(s) = \max_{\pi} \mathbb{E}[r_{\pi t} + Q_{t+1}(s,\pi)] \tag{3}$$

If the problem space is small enough then the set of policies and state can be fully observed and Equation (3) can be used to determine the optimal policy toward the target. However, in most cases, the problem space is too complex to explore, and hence evaluating the future reward for all possible policies is not feasible. $Q_t^*(s)$ is therefore approximated by a non-linear deep network $Q^*(s,\theta)$ with a set of parameters θ resulting in what is known as deep Q-learning (34).

2.1.2. Agent state, action definition, and reward function

Given a 3-D scan as the agent's environment, a state s is represented by (s_x, s_y, s_z) which is the top-left corner of a $(64 \times 64 \times 64)$ cube contained in the 3-D scan. We adopt an agent history approach which involves feeding the last four states visited by the agent to the network to prevent the agent from getting stuck in a loop. Since we have a fixed-sized cube as a state our agent's set of six actions $\{m_u, m_d, m_l, m_r, m_f, m_b\}$ is made up of only movements up, down, left, right, forward, and backward respectively which enables the agent to visit all possible locations within the volume. The agent's reward for taking an action a is a function of the intersection over union (IoU) of the target state s^* and the state before (s_{ab}), and after (s_{aa}) taking the action. This is given by

$$R_a(s_{aa}, s_{ab}) = sign[IoU(s_{aa}, s^*) - IoU(s_{ab}, s^*)] \tag{4}$$

where *sign* is the sign function that returns -1 for all values less than 0 and 1 otherwise. This leads to a binary reward ($r \in \{-1, 1\}$) scheme which represents good and bad decisions respectively. During the training stage, the agent search sequence is terminated when the IoU of the current agent's state and the target state is

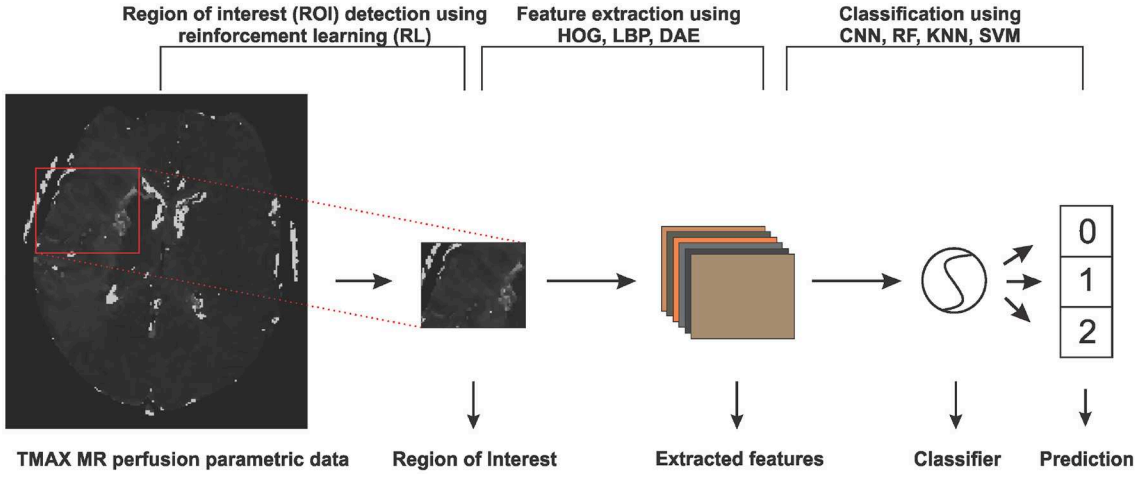


FIGURE 1
An overview of the steps involved in predicting collateral flow grading from MR perfusion parametric data. The first step involves a region of interest detection using reinforcement learning, followed by histogram of gradient (HOG), local binary pattern (LBP), and denoising autoencoder (DEA) feature extraction schemes and then the classification step which uses random forest (RF), K-nearest neighbor (KNN), support vector machine (SVM), and convolutional neural network (CNN) classifiers.

greater than or equal to a predefined threshold τ . At test time the agent is terminated when a sequence of decisions leads to an oscillation [as proposed by Alansary et al. (33)], that is when the agent visits one state back and forth for a period of time.

Experiments by Alansary et al. (33) and Navarro et al. (35) show that deep reinforcement learning has superior performance in object detection as compared to classical supervised learning, especially in images with a noisy background. RL agents also require lesser training data as compared to other supervised learning methods like CNN. These proven advantages make deep reinforcement learning the right choice for our limited and noisy data.

2.2. Feature extraction and classification

Feature extraction methods are used in many machine learning tasks to either reduce the dimension of the problem or to extract information from the raw input which would otherwise not be easily extracted by the underlying classifier. In this work, we extract two main classes of features—learned features through a denoising auto-encoder (DAE), and local image descriptors made up of histogram of oriented gradients (HOG) and local binary pattern (LBP).

2.2.1. Denoising auto-encoder

An auto-encoder is an unsupervised deep learning method used for dimension reduction, feature extraction, image reconstruction or denoising and is sometimes also used as a pre-training strategy in supervised learning networks. An auto-encoder is made up of two parts: an encoder $\Phi : \mathcal{X} \rightarrow \mathcal{F}$ which maps an image $x \in X$ to $f_x \in \mathcal{F}$ in the features domain and a decoder $\Psi : \mathcal{F} \rightarrow \mathcal{X}$ which maps a feature set $f \in \mathcal{F}$ to $x_f \in X$. The full auto-encoder is therefore a composite function of the form $\Psi \circ \Phi : \mathcal{X} \rightarrow \mathcal{X}$. Let $\widehat{y} = \Psi(\Phi(x))$ for a given input image $x \in \mathcal{X}$, then the learning process of auto-encoder involves finding a pair of $\{\Phi, \Psi\}$ such that $\widehat{y}_i = x_i$ for all $x_i \in \mathcal{X}$. The encoder Φ then becomes the feature extractor which is used for extracting the needed features.

If the function Φ is invertible, then the learning process can lead to a trivial solution by just choosing Ψ to be the inverse of Φ, and $\Psi \circ \Phi$ becomes an identity function leading to what is known as identity-function risk. To prevent this, the input image x is first corrupted by adding noise before feeding it to Φ leading to a denoising auto-encoder. We therefore have

$$\widehat{y} = \Psi(\Phi(\tilde{x})), \quad \tilde{x} = \Upsilon(x) \tag{5}$$

where Υ is the random image corruption function. We approximate the encoder and decoder by deep CNNs $E(x, \theta_e)$ and $D(f, \theta_d)$ parameterized by θ_e and θ_d; respectively. Training is done through back-propagating the Mean Squared Error (MSE) of the original image x and the reconstructed image $\widehat{y}$ given by

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (\widehat{y}_i - x_i)^2 \tag{6}$$

where N is the number of images in the training set or training batch. We adopt the V-Net architecture proposed by Milletari et al. (36) and simplify it by removing the fine-grained feature forwarding, and reducing the depth of the network due to limitations on the amount of training data available. The downsampling layers of the VNET architecture represent the encoding part $[E(x, \theta_e)]$ of the DAE and the upsampling layers represent the decoding part $[D(f, \theta_d)]$ of the DAE. Figure 2 shows an overview of the simplified architecture used for extracting the DAE features.

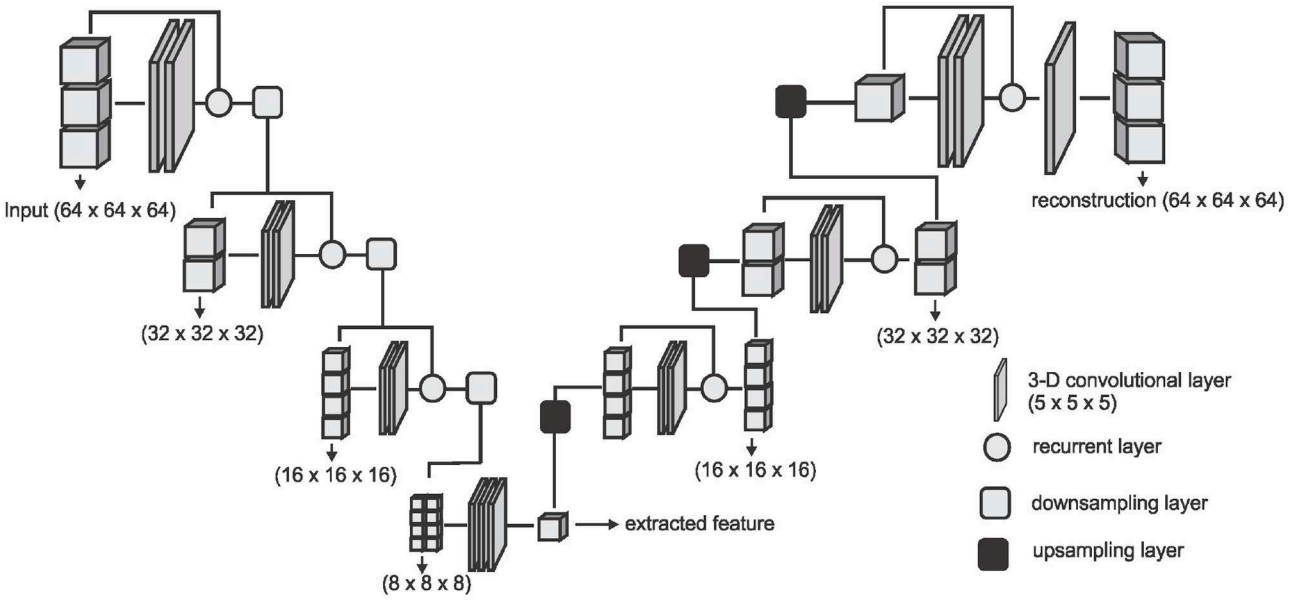


FIGURE 2
The network architecture used for extracting the DAE features. The downsampling layer is a convolution with a stride of (2 × 2 × 2) which downsamples the input volume to half of the size on every axes. The upsampling layer is a transposed convolution with a stride of (2 × 2 × 2) which doubles the size of the input on every axes.

2.2.2. Local image descriptors and classifiers

We consider two types of local image descriptors - histograms of oriented gradients (HOG) and a local binary pattern (LBP). Given a volume X, we extract the LBP encoding of each voxel by thresholding its $3 \times 3 \times 3$ neighborhood by the intensity value p^* of the center voxel which results in 26 long bits $b_0, b_1, b_2, ..., b_{25}$ where $b_i = \{1, \text{if } p_i \geq p^*, 0 \text{ otherwise}\}$ and p_i is the intensity value of the ith neighbor. We then concatenate the binary encoding to a single binary number $b_0 b_1 b_2 ... b_{25}$ and then into a decimal value which results in 2^{25} possible binary codes. Details of the implementation until this point can be found in Heikklä and Pietikäinen (37). We group the codes into two main classes—uniform codes which have at most two binary transitions and non-uniform codes which have more than two binary transitions. A binary transition is a switch from 0 to 1 or vice versa. For example the codes 0000, 000111, 011100, and 110110 have zero, one, two, and three transitions respectively. To handle noisy data and to reduce the feature space, we group all the non-uniform codes into one class and add it to the uniform codes resulting in 927 codes instead of 2^{25} . Finally, the histogram distribution of the individual codes is extracted as the LBP feature representation for the volume X.

We also explore the HOG feature extractor based on the method proposed in Klaser et al. (38). Given a volume X, we quantize gradient orientations over an icosahedron and merge opposite directions in one bin resulting in 10 gradient orientations. The gradient for each voxel $x_i \in X$ is obtained by convolving the $5 \times 5 \times 5$ neighborhood of the voxel by gradient filters k_x , k_y , and k_z of the same size, giving us a gradient vector $\vec{x}_i \in \mathbb{R}^3$. The gradient filters are zero everywhere except for the center columns along the respective axes $k_x(i, 3, 3) = k_y(3, i, 3) = k_z(3, 3, i) = [1, 0, -2, 0, 1]$ for $i \in \{1, 2, ..., 5\}$. The gradient vectors $\vec{x}_i$ are then projected to the gradient orientations and a histogram representation of these orientations are obtained and used as the HOG feature representation of the volume X.

We run experiments with four machine learning classifiers on each of the features extracted. We implement Convolutional Neural Network (CNN), Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN) classifiers. Our CNN classifier in Figure 3 has four convolutional layers, aimed at extracting local image features, followed by two fully connected layers and a sigmoid layer for classification. Each layer is followed by a non-linear hyperbolic tangent (tanh) activation function. For classification based on the HOG, LBP, and DEA features, we remove the convolutional layers and feed the features directly to the fully connected layers and then the sigmoid layer for the classification. For the RF, SVM, and KNN classifiers we use the implementation of these classifiers from the Scikit-Learn library (39) in python.

3. Experiments and results

3.1. Patient population and image data

We test our proposed methods on parametric volumes extracted from MR perfusion data from 183 patients with acute ischemic stroke. Details of the image acquisition and preparation are already published by Pinto et al. (40). Our dataset is made up of three parametric information—T_{max} volumes which refer to the time taken for the blood flow to reach its peak, relative blood flow (rBF) volumes which refer to the volume of blood passing through a given brain tissue per unit of time, and relative blood volume (rBV) defined as the volume of blood in a given brain tissue relative to an internal control (e.g., normal white matter or an arterial input function). Each volume has a resolution of (0.9, 0.9, and 6.5 mm)

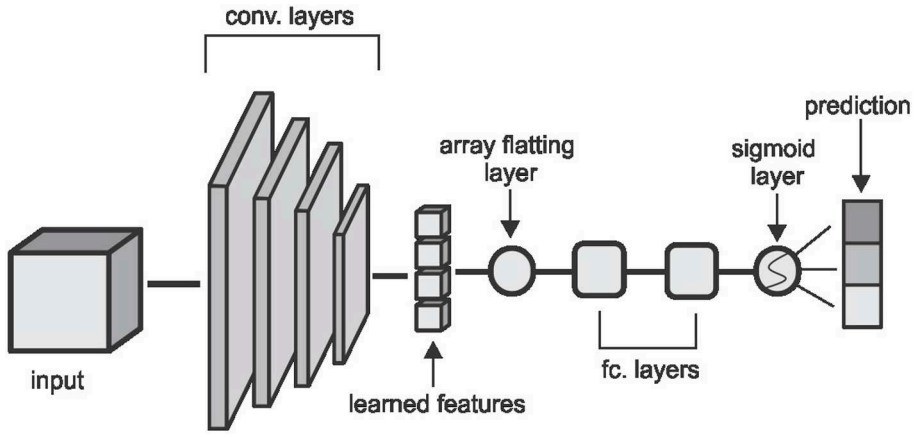


FIGURE 3
The CNN architecture used in the classification task. Convolutional layers are made up of (5 × 5 × 5) kernels with a stride of (2 × 2 × 2) which reduces the volume by half of the input size on each layer. The first two layers extract 2 feature cubes and the last two layers extract 4 feature cubes each. The fully connected layers have 64 and 32 hidden nodes respectively and the convolutional and fully connected layers are followed by a non-linear hyperbolic tangent (tanh) activation function.

and a dimension of (256, 256, and 19) voxels on the sagittal, coronal and axial planes respectively. Ground truth labels are obtained from a trained neuroradiologist, with over ten years of experience, who manually investigates the DSA slides of the associated patient and assigns one of three labels (0-poor, 1-medium, 2-good) to this patient. We use these labels for a 3-class prediction experiment and we also experiment on a risk-stratified nested test where we first predict good - (2) against not good (0, 1) collaterals and then separate the not good class into poor (0) and medium (1) collaterals in a cascaded approach.

TABLE 1 Quantitative results from the region of interest detection task.

Type	Class	Mean	Std	Max	Min
IoU	0	0.49	0.22	0.79	0.08
	1	0.52	0.14	0.81	0.09
	2	0.42	0.21	0.81	0.04
Center points displacement (in voxels)	0	20	13	51	5
	1	17	9	52	4
	2	23	14	63	5

IoU refers to the intersection over union ratio between the prediction and the ground truth. Center point displacement is the euclidean distance between the predicted center point and the ground truth center point.

3.2. Preprocessing

Our image preprocessing involves two main tasks. First, we make our datasets isotropic by applying a B-spline interpolation to the axial axis since the other two axes have the same spacing leading to volume with a resolution of 0.9 mm on each plane and a new dimension of (256, 256, and 127). This is followed by an extraction of the brain region from the skull using the brain extraction tool (BET) from the ANTS library. The brain extraction is carried out on the T_{max} volumes and the resulting mask is then applied to the rBF and rBV volumes.

3.3. Region of interest localization

After the preprocessing step we extract the occluded regions as the region of interest (ROI) using the reinforcement learning architecture described in Section 2.1. We adopt the network architecture from Alansary et al. (33) with modifications proposed in Navarro et al. (35). A stopping criterion of $\tau = 0.85$ is used during training—that is, an intersection over union (IoU) value greater than or equal to 0.85 implies that the region of interest is detected. We perform the ROI detection task on the T_{max} volumes since the occluded regions are easier to detect in these volumes. The resulting cube region is then applied on the rBF and rBV volumes to extract the corresponding cubes in these volumes as well. For each volume, we select 20 starting cubes of size (64 × 64 × 64) at random and run the agent till the stopping criterion is reached. We then aggregate the results from the 20 different runs to get the prediction of the final ROI. After getting the region we extract the mirror of the ROI (ROI+M) by reflecting the ROI on the opposite side of the brain and using it as an additional feature. This results in 6 cubes per patient (i.e., two volumes each from T_{max} , rBF, and rBV volumes). Qualitative and quantitative results from the region of interest extraction can be found in Table 1 and Figures 4, 5. From the box plots in Figure 5, it is evident that the region of interest detection was more successful in the poor collateral flow classes (class 0 and 1) than in the good collateral flow class. This can be explained by the fact that in cases of good collateral flow, there is a uniform distribution of the T_{max} value within the occluded region and its neighborhood making it hard for the RL agent to detect the ROI. From Figure 4 we observe that in most cases the ground truth does not cover the total occluded region [e.g., column (b)] and hence the predicted ROI, though does not completely overlap the ground truth, still contains other parts of the occluded region which

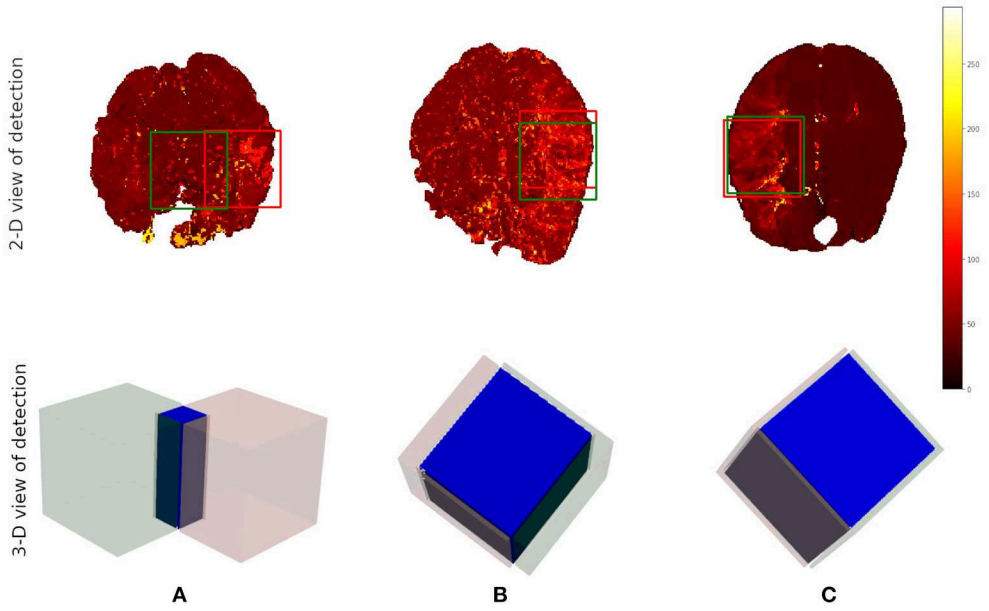


FIGURE 4
Qualitative results from the ROI detection task. The top row is the axial view of the ground truth (in red) and the prediction (in green). The bottom row is a 3-D visualization of the ground truth cube (in red), the predicted cube (in green), and the intersection between the two (in blue). Column **(A)** corresponds to the worst prediction in our test set while column **(C)** refers to the best result in terms of IoU. In column **(B)**, we can observe that though the overlap is not perfect the prediction still contains some part of the occluded region which is not in the ground truth. This implies that though we have poor scores we still have good ROI detection which can be used for the classification task.

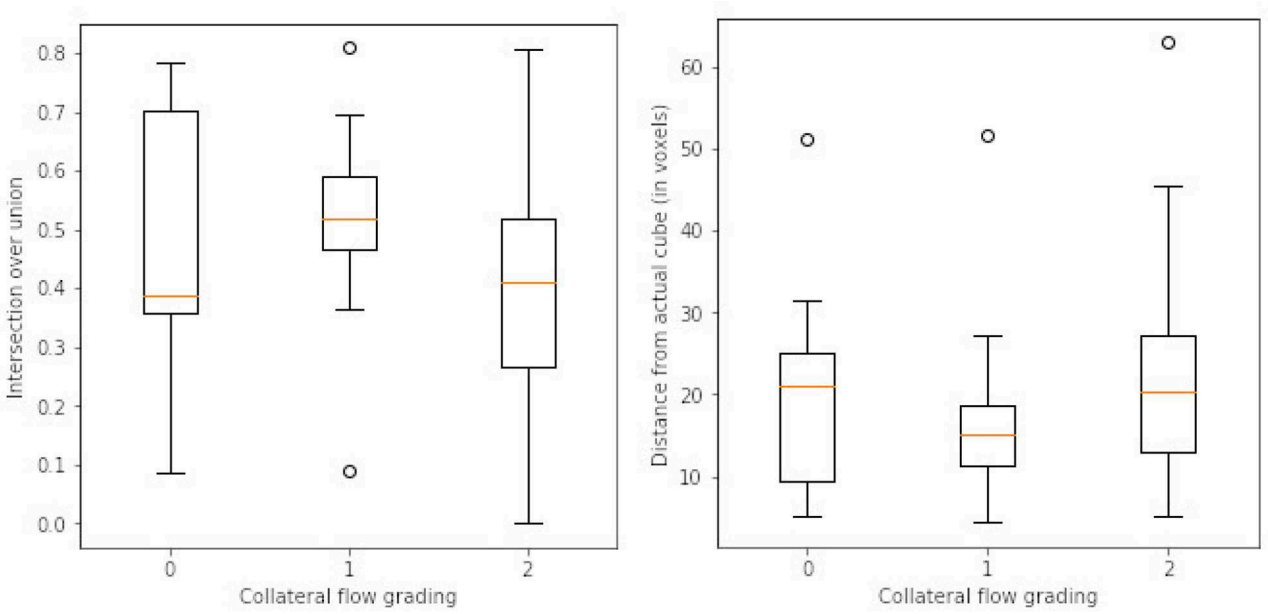


FIGURE 5
Box plots of results from ROI detection task. **Left** is the intersection over union (IoU) ratio between the prediction and the ground truth over the three classes. **Right** is the euclidean distance between the predicted center point and the ground truth center point. From the distributions, it is clear that it is easy to detect the ROI in the poor collateral flow class (class 0) compared to the good collateral flow class (class 2). This can be explained by the fact that in good collateral flow cases T_{max} shows uniform values in the whole volume.

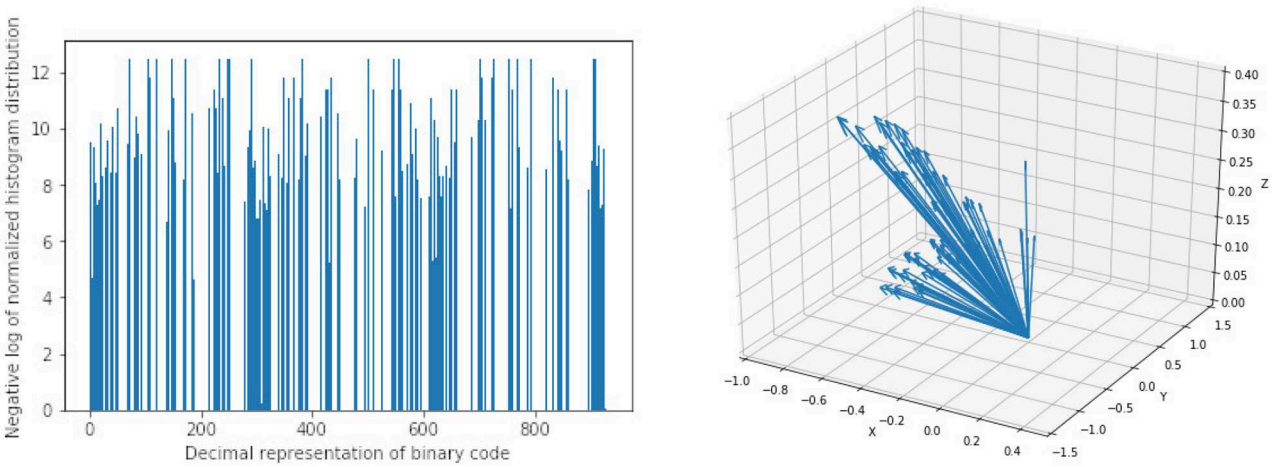


FIGURE 6
Sample of extracted features using Local Binary Pattern (LBP) on the **left** and Histogram of Oriented Gradients (HOG) on the **right**. Bar heights from LBP features represent the frequency of a given pattern and the position on the x-axis is the decimal representation of the binary pattern. Arrow directions in HOG features are the gradient vectors and the length of the arrows represents the frequency of the given gradient. The LBP features show a uniform distribution of the extracted patterns with no dominant pattern. The HOG features on the other hand show evidence of high gradients in the extracted region of interest.

TABLE 2 **Results from preliminary experiments on collateral flow grading.**

Type	Method	RAW	ROI	ROI+M	DAE	HOG	LBP
Three classes	CNN+MLP	0.51(±0.04)	0.63(±0.06)	0.65(±0.03)	0.50(±0.07)	0.38(±0.13)	0.25(±0.14)
	RF	0.51(±0.02)	0.65(±0.04)	0.67(±0.05)	0.66(±0.04)	0.69(±0.02)	0.60(±0.05)
	KNN	0.48(±0.10)	0.54(±0.02)	0.58(±0.05)	0.55(±0.06)	0.59(±0.02)	0.43(±0.04)
	SVM	0.56(±0.04)	0.66(±0.05)	**0.70(± 0.03)**	0.70(±0.04)	0.53(±0.02)	0.25(±0.15)
Cascaded (two step)	CNN+MLP	0.55(±0.01)	**0.72(± 0.05)**	0.70(±0.04)	0.66(±0.05)	0.54(±0.02)	0.21(±0.13)
	RF	0.47(±0.07)	0.67(±0.03)	0.64(±0.03)	0.65(±0.04)	0.70(±0.03)	0.56(±0.07)
	KNN	0.44(±0.04)	0.55(±0.06)	0.56(±0.07)	0.52(±0.08)	0.60(±0.05)	0.48(±0.08)
	SVM	0.38(±0.02)	0.51(±0.04)	0.46(±0.04)	0.51(±0.04)	0.46(±0.04)	0.10(±0.00)

RAW features refer to the full-sized three parametric volumes (T_{max} , rBF, and rBV) after skullstripping. ROI refers to the corresponding cubes extracted from the parametric volumes based on the manually annotated ROI and ROI+M is the ROI combined with its mirror cube on the opposite side of the brain. Other features (DEA, HOG, and LBP) are all extracted from the ROI cubes. Scores represent mean accuracy over the 5-fold cross-validation experiments with their corresponding standard deviation in parenthesis.
Values in bold refer to the feature-classifier combination with the highest accuracy under each experiment type.

is not captured in the ground truth and it is therefore sufficiently accurate for the classification task.

3.4. Classification

3.4.1. Feature representations

In total three sets of features (DAE, HOG, and LBP) are extracted in addition to the actual extracted cube (ROI) and its mirror cube (ROI+M). We learn features automatically through an unsupervised denoising auto-encoder. The network takes the extracted ROI cubes from the T_{max}, rBV, and rBF volumes as three input channels and produces a single channel feature set of size ($8 \times 8 \times 8$). We normalize the cubes individually into the range [0, 1] before feeding them to the network.

For HOG features we extract 10 features each for the three parametric volumes and concatenate them into a vector of length 30 for the classification task. Figure 6 shows a sample of the extracted HOG features for a patient for the three input channels. Finally, LBP features are extracted using the method described in Section 2.2. Here we combine all the three channels and run the histogram over the three channels which results in a 927 feature vector as explained in Section 2.2. Figure 6 shows a sample of the extracted LBP features from our dataset.

3.4.2. Classifier training

We handle the collateral flow classification through two main approaches—a three-class multi-label classification task where we predict three labels in one step, and a two-step cascaded approach where we predict a binary label of classes (0, 1) against 2 in the first step and separate the class 0 from 1 in the second step. We implement our CNN architecture using the Keras library (41) in python with TensorFlow as the backend. Random forest, support vector machine, and K-nearest neighbor classifiers were

implemented using the scikit-learn library (39) in python. We set up our experiments as follows:

CNN classifier: For the CNN classifier, we use a weighted categorical cross-entropy with a weight of $\frac{1}{|k|}$ for each class k in the training set. A stochastic gradient descent optimizer with a learning rate of 0.001, decay of $1e^{-6}$, and momentum of 0.9 is used to fine-tune the network parameters at 20 epochs.

K-Nearest neighbor classifier: We conduct preliminary a experiment with a grid search to know which parameters will work best. For our final experiment, we use $k = 3$ neighbors with uniform weights, a leaf size of 30, and the Minkowski metric.

Random forest classifier: After the initial grid search experiment, we implement the classifier with 200 estimators, and the Gini impurity function is used to measure the quality of a split.

Support vector machine classifier: We use a regularization parameter $C = 10$, a third-degree polynomial kernel, a balanced class weight, and a tolerance of $1e^{-3}$ for the stopping criterion.

3.4.3. Classification results

We test different combinations of the feature sets extracted in the previous experiments and classifiers discussed in a preliminary experiment and present the results in Table 2. Due to limitations in the size of the dataset, we adopt a 5-fold cross-validation approach in a preliminary experiment instead of a training-validation-test splitting approach and report the average scores over the accuracy in the individual validations. In the preliminary experiments (results in Table 2), we use the manually annotated ROI and not the ROI predicted from the proposed reinforcement learning. We later, in a follow-up experiment, compare the performance of the proposed CNN on manually annotated ROI and the predicted ROI (results in Table 3).

The results in Table 2 show that the region of interest extraction step helps improve the results in all classification methods. This can be verified by comparing the performance from the full image (RAW column) with the performance of the region of interest (ROI column) in Table 2. Also by adding the mirror of the occluded region to the extracted ROI (ROI+M) we achieve improved results in most of the classifiers with performance falling in classifiers like KNN and SVM due to the increase in the dimension of data introduced by the mirror of the ROI. The cascaded method shows higher accuracy in almost all the classifier-feature combinations when compared to the direct three-class prediction. This can be explained by the distribution of classes in the dataset. That is, for the cascaded approach we have fairly balanced data when we combine poor and moderate flow against good collateral flow which is not the case with the direct three-class multi-label prediction approach. It, therefore, suggests that in cases where we have highly imbalanced class distributions a multi-label classification might perform poorly. The overall performance of CNN is better than the other machine learning classifiers and can be explained by the fact that the convolutional layers of the CNN architecture extract features while paying attention to the class of the input data. This makes the feature extraction process more efficient than the other feature extraction schemes which have no knowledge of the underlying label of the input data at the time of extracting the features. Again CNN with only ROI data performs slightly better than with the mirror of the ROI (72 vs. 70% in Table 2) and this can also be explained by the fact that the CNN used in our experiments is fairly shallow and hence could not handle the additional feature dimensions introduced by the mirrored images.

TABLE 3 Results from the experiment on collateral flow grading using only ROI data on our proposed cascaded CNN.

Input data	Binary	Three classes
Manual ROI	0.84	0.74
Automated ROI	0.80	0.72

Manual ROI refers to the ground truth ROI and automated ROI refers to the predicted ROI from our proposed Reinforcement Learning approach. Binary refers to the result from the first binary classification (i.e., {0,1} vs. 2) and three classes is the three-class classification based on the cascaded networks.

Based on the results of the preliminary experiment, we further probe into the training of the proposed CNN classifier with the ROI data. In this experiment, we split the data into training and testing sets. The test set is made up of 50 volumes randomly selected with reference to the ratio of class count in the entire dataset. We make use of both the manually annotated ROI and the automated ROI from our proposed Reinforcement Learning approach during training. We finally evaluate the trained models on the automated ROI and compare it with the same network trained and evaluated solely on the manually annotated ROI data. Table 3 shows the result of this experiment.

The results in Table 3 from our follow-up experiment show that the automated ROI from the proposed Reinforcement Learning approach is comparable to the manually detected ROI in terms of predicting collateral flow (2% drop in accuracy which represents one out of the 50 patients in the test set). This is crucial in automating the whole collateral flow prediction workflow in a clinical setting.

4. Summary and conclusion

In this work, we present a deep learning approach toward grading collateral flow in ischemic stroke patients based on parametric information extracted from MR perfusion data. We start by extracting regions of interest using deep reinforcement learning. We then learn denoising auto-encoder features and modern implementation of 3-D HOG and LBP features. We proceed to the actual classification task using a combination of the extracted features and CNN, random forest, K-nearest neighbor, and support vector machine classifiers.

Our experiments show that the rich information on blood flow visible from MRI perfusion can be used to predict collateral flow in a similar manner to DSA images which are invasive in nature. Region of interest detection with reinforcement learning is successful to an acceptable level and can be used as a guide to estimate the region in the brain which requires more attention. It is evident that high class imbalance can be a major challenge in the collateral flow grading task and many similar works. We however show that for datasets with high class imbalance, a two-step cascaded classification approach performs better than a one-time multi-label classification method. It is also evident from our results that a direct CNN classifier is able to extract relevant features

from the region of interest and has an advantage over classical machine learning classifiers like RF, KNN, and SVM that depend on handcrafted features like HOG and LBP.

Collateral flow grading is an essential clinical procedure in the treatment of ischemic stroke patients. We have presented a framework for automating the process in clinical setup and have achieved promising results given our limited dataset. For the proposed framework to be clinically useful there is the need for further tests with possibly more data from multiple stroke centers. The grading can also be customized for specific patient groups for example providing information about age group, gender, and other biographical and historical information of patients as an additional feature can help improve the result of the framework.

Data availability statement

Data is currently private, pending approval from the source of the data. Requests to access these datasets should be directed to giles.tetteh@tum.de.

Author contributions

GT and BM contributed to the conception and design of the study. GT, FN, and JP contributed to the experiments and computer codes. RM, JK, and RW organized the database. GT wrote the first draft of the manuscript. BM, JP, and JK contributed to the final manuscript. BM, CZ, JK, and RW handled the administrative tasks. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

GT and BM acknowledge support by the Deutsche Forschungsgemeinschatf (DFG) (ME 3511/5-1).

Acknowledgments

BM acknowledges support from the Helmut-Horten-Foundation.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Faber J, Chilian W, Deindl E, van Royen N, Simons M. A brief etymology of the collateral circulation. *Arterioscler Thromb Vasc Biol*. (2014) 9:1854–9. doi: 10.1161/ATVBAHA.114.303929

2. Liu L, Go T, Leng X, Pu Y, Huang LA, Xu A, et al. Guidelines for evaluation and management of cerebral collateral circulation in ischaemic stroke 2017. *Stroke Vasc Neurol*. (2018) 3:svn-2017. doi: 10.1136/svn-2017-000135

3. Ginsberg M. The cerebral collateral circulation: Relevance to pathophysiology and treatment of stroke. *Neuropharmacology*. (2017) 134:280–92. doi: 10.1016/j.neuropharm.2017.08.003

4. Ben Hassen W, Malley C, Boulouis G, Claren?on F, Bartolini B, Bourcier R, et al. Inter- and intraobserver reliability for angiographic leptomeningeal collateral flow assessment by the American Society of Interventional and Therapeutic Neuroradiology/Society of Interventional Radiology (ASITN/SIR) scale. *J Neurointerv Surg*. (2018) 11:338–41. doi: 10.1136/neurintsurg-2018-014185

5. McVerry F, Liebeskind D, Muir K. Systematic review of methods for assessing leptomeningeal collateral flow. *AJNR Am J Neuroradiol*. (2011) 33:576–82. doi: 10.3174/ajnr.A2794

6. Martinon E, Lefevre P, Thouant P, Osseby G, Ricolfi F, Chavent A. Collateral circulation in acute stroke: Assessing methods and impact: A literature review. *J Neuroradiol*. (2014) 41:97–107. doi: 10.1016/j.neurad.2014.02.001

7. Sheth SA, Liebeskind DS. Imaging evaluation of collaterals in the brain: physiology and clinical translation. *Curr Radiol Rep*. (2014) 29:1–9. doi: 10.1007/s40134-013-0029-5

8. Jansen IGH, Berkhemer OA, Yoo AJ, Vos JA, Lycklama A, Nijeholt GJ, et al. Comparison of CTA- and DSA-based collateral flow assessment in patients with anterior circulation stroke. *AJNR Am J Neuroradiol*. (2016) 37:2037–42. doi: 10.3174/ajnr.A4878

9. Muehlen I, Kloska SP, Gölitz P, Hölter P, Breuer L, Ditt H, et al. Noninvasive collateral flow velocity imaging in acute ischemic stroke: intraindividual comparison of 4D-CT angiography with digital subtraction angiography. In: *RöFo-Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren*. Georg Thieme Verlag KG. (2019). Vol. 191, p. 827-835. doi: 10.1055/a-0825-6660

10. Kauw F, Dankbaar JW, Martin BW, Ding VY, Boothroyd DB, van Ommen F, et al. Collateral status in ischemic stroke: a comparison of computed tomography angiography, computed tomography perfusion, and digital subtraction angiography. *J Comput Assist Tomogr*. (2020) 44:984–92. doi: 10.1097/RCT.0000000000001090

11. Jung S, Wiest R, Gralla J, McKinley R, Mattle H, Liebeskind D. Relevance of the cerebral collateral circulation in ischaemic stroke: Time is brain, but collaterals set the pace. *Swiss Med Wkly*. (2017) 147:w14538. doi: 10.4414/smw.2017.14538

12. Maija R. *Collateral blood supply in acute stroke*. (2015). doi: 10.1594/ecr2015/C-1460

13. Bang OY, Saver J, Kim S, Kim GM, Chung CS, Ovbiagele B, et al. Collateral flow predicts response to endovascular therapy for acute ischemic stroke. *Stroke*. (2011) 42:693–9. doi: 10.1161/STROKEAHA.110.595256

14. Bang OY, Saver J, Buck B, Alger JR, Starkman S, Ovbiagele B, et al. Impact of collateral flow on tissue fate in acute ischaemic stroke. *J Neurol Neurosurg Psychiatry*. (2008) 79:625–9. doi: 10.1136/jnnp.2007.132100

15. Mansour O. Assessment of collateral pathways in acute ischemic cerebrovascular stroke using a mansour grading scale; a new scale, a pilot study. *Internet J Internal Med*. (2013) 3:1–10.

16. Maas M, Lev M, Ay H, Singhal A, Greer D, Smith W, et al. Collateral vessels on CT angiography predict outcome in acute ischemic stroke. *Stroke*. (2009) 40:3001–5. doi: 10.1161/STROKEAHA.109.552513

17. Tan I, Demchuk A, Hopyan J, Zhang L, Gladstone D, Wong K, et al. CT angiography clot burden score and collateral score: correlation with clinical and radiologic outcomes in acute middle cerebral artery infarct. *AJNR Am J Neuroradiol*. (2009) 30:525–31. doi: 10.3174/ajnr.A1408

18. McHugh ML. Interrater reliability: the kappa statistic. *Croatian Soc Med Biochem Lab Med*. (2012) 22:276–82. doi: 10.11613/BM.2012.031

19. Kersten-Oertal M, Alamer A, Fonov V, Lo BWY, Tampieri D, Collin DL. Towards a computed collateral circulation score in ischemic stroke. *arXiv [Preprint]*. (2020). arXiv: 2001.07169. doi: 10.48550/ARXIV.2001.07169

20. Grunwald IQ, Kulikovski J, Reith W, Gerry S, Namias R, Politi M, et al. Collateral automation for triage in stroke: evaluating automated scoring of collaterals in acute stroke on computed tomography scans. *Cerebrov Dis*. (2019) 47:1–6. doi: 10.1159/000500076

21. Pauli J. *Learning-Based Robot Vision, Principles and Applications*. Berlin: Springer Science & Business Media. (2001). doi: 10.1007/3-540-45124-2

22. DeSouza G, Kak A. Vision for mobile robot navigation: a survey. *IEEE Trans Pattern Anal Mach Intell*. (2002) 24:237–67. doi: 10.1109/34.982903

23. Bonin-Font F, Ortiz A, Oliver G. Visual navigation for mobile robots: a survey. *J Intell Robot Syst*. (2008) 53:263–96. doi: 10.1007/s10846-008-9235-4

24. Peng J, Bhanu B. Delayed reinforcement learning for closed-loop object recognition. In: *Proceedings of 13th International Conference on Pattern Recognition*. IEEE. (1996). p. 310–314.

25. Peng J, Bhanu B. Closed-loop object recognition using reinforcement learning. *IEEE Trans Pattern Anal Mach Intell*. (1998) 20:139–54. doi: 10.1109/34.659932

26. Taylor G, Wolf C. Reinforcement learning for parameter control of text detection in images from video sequences. In: *Proceedings 2004 International Conference on Information and Communication Technologies: From Theory to Applications*. (2004). p. 517–518.

27. Sahba F, Tizhoosh HR, Salama MM. A reinforcement learning framework for medical image segmentation. In: *The 2006 IEEE international joint conference on neural network proceedings*. IEEE (2006). p. 511–517. doi: 10.1109/IJCNN.2006.246725

28. Sahba F, Tizhoosh HR, Salama MM. Application of opposition-based reinforcement learning in image segmentation. In: *2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing*. IEEE (2007). p. 246–251. doi: 10.1109/CIISP.2007.369176

29. Shokri M, Tizhoosh HR. Using reinforcement learning for image thresholding. In: *CCECE 2003-Canadian Conference on Electrical and Computer Engineering*. IEEE (2003). p. 1231–1234.

30. Sahba F, Tizhoosh HR, Salama MM. Using reinforcement learning for filter fusion in image enhancement. In: *IASTED International Conference on Computational Intelligence*. (2005). p. 262–266.

31. Tizhoosh H, Taylor G. Reinforced contrast adaptation. *Int J Image Graphics*. (2006) 6:377–92. doi: 10.1142/S0219467806002379

32. Netto SM, Leite VR, Silva AC, de Paiva AC, de Almeida Neto A. Application on reinforcement learning for diagnosis based on medical image. In: *Reinforcement Learning*. IntechOpen (2008) 379.

33. Alansary A, Oktay O, Li Y, Folgoc L, Hou B, Vaillant G, et al. Evaluating reinforcement learning agents for anatomical landmark detection. *Med Image Anal*. (2019) 53:156–64. doi: 10.1016/j.media.2019.02.007

34. Mnih V, Kavukcuoglu K, Silver D, Rusu A, Veness J, Bellemare M, et al. Human-level control through deep reinforcement learning. *Nature*. (2015) 518:529–33. doi: 10.1038/nature14236

35. Navarro F, Sekuboyina A, Waldmannstetter D, Peekan JC, Combs SE, Menze BH. Deep Reinforcement Learning for Organ Localization in CT. *Proc Mach Learn Res*. (2020) 121:544–54.

36. Milletari F, Navab N, Ahmadi SA. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 *Fourth International Conference on 3D vision (3DV)*. IEEE (2016). p. 565–571. doi: 10.1109/3DV.2016.79

37. Heikklä M, Pietikäinen M. A texture-based method for modeling the background and detection moving objects. *IEEE Trans Pattern Anal Mach Intell*. (2006) 28:657–62. doi: 10.1109/TPAMI.2006.68

38. Klaser A, Marszaäek M, Schmid C. A spatio-temporal descriptor based on 3d-gradients. In: *BMVC 2008-19th British Machine Vision Conference*. British Machine Vision Association (2008). p. 275–1. doi: 10.5244/C.22.99

39. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in python. *J Mach Learn Res*. (2011) 12:2825–30.

40. Pinto A, Amorim J, Hakim A, Alves V, Reyes M, Silva CA. Prediction of stroke lesion at 90-day follow-up by fusing raw DSC-MRI with parametric maps using deep learning. *IEEE Access*. (2021) 9:26260–70. doi: 10.1109/ACCESS.2021.3058297

41. Chollet F. Keras: GitHub (2015). Available online at: https://github.com/fchollet/keras (accessed August 31, 2022).

Part III

DISCUSSION AND OUTLOOK

6

DISCUSSION

This research aims to present methodological advances toward the transition of deep learning from natural image processing to medical image analysis in general. Specifically, this thesis focuses on addressing some of the challenges that arise from the application of deep learning methods to the analysis of vascular and curvilinear structures. Drawing inspiration from the success stories in natural image processing, several variants of convolutional neural networks (CNNs) have been adopted for the purpose of medical image processing and hence the methodological framework of this dissertation is centered around CNNs.

This publication-based dissertation is made up of three first-author conference and journal publications in Part 2 (ii) which outlines the methodological contributions of this work and three co-author conference and journal publications in Appendix A which cover the application of the proposed methodologies and serve as a further proof of the research findings.

The work starts with the extraction of deep features using a multi-scale convolutional neural network. We explore the idea of inception architecture proposed by Szegedy et al. [48] which is based on finding out how an optimal local sparse structure in a convolutional vision network can be approximated and covered by readily available dense components. By replacing convolutional operations with mini-networks inception models reduce the number of parameters in a CNN and lead to fewer computations. The publication in Chapter 3 presents the feature extraction scheme and its application to the publicly available Digital Retinal Images for Vessel Extraction (DRIVE) dataset [47] for the tasks of vessel segmentation and centerline prediction. The results in Chapter 3 show that inception models can be used to reduce network parameter size and computational requirement and at the same time achieve comparable segmentation and prediction accuracy.

Chapter 4 presents the next and major study in this dissertation. Here we present a careful study of some challenges associated

with applying deep CNNs to medical images in general and specifically to the analysis of vascular networks. The following three main challenges are identified and several methodological contributions are proposed to address these challenges as part of the deepvesselnet study [49].

HIGH COMPUTATION AND MEMORY REQUIREMENTS Unlike natural images, medical images are normally 3-D and of high resolution. Processing of these images involves 3-D convolutional operations which increases memory and computational needs exponentially. The early solution has been to use 2-D based networks which leads to the loss of 3-D context information and impacts the accuracy of the learning process. In Chapter 4 we propose cross-hair filters as a replacement for the classical 3-D convolutional operation. We show mathematically and through experiments on actual medical data that cross-hair filter reduces computational and memory burden introduced by classical 3-D convolutional operations. At the same time, we show that cross-hair filters achieve comparable accuracy as classical 3-D filters. In a follow-up experiment by Todorov et al. [52] in Appendix A.3, the proposed cross-hair filters are used in a deep network to segment the vascular network in the whole mouse brain dataset. The results from the above-mentioned work confirm the effect of the proposed cross-hair filters on computational and memory requirement and show a superior performance compared to classical 3-D filters.

HIGH CLASS IMBALANCE In most medical image processing tasks the object of interest accounts for less than 5% of the entire data points. This leads to an imbalance between the distribution of the data in the class of the object of interest (foreground) and the background or irrelevant data points. Our study showed that the problem is significantly worse for vessel segmentation, centerline prediction, and bifurcation detection tasks where vessels account for less than 2.5%, and vessel centerlines and bifurcation points account for less than 0.3%. This imbalance in data leads to bias in the learning process for deep networks. The obvious solution to this problem has been the introduction of a class balancing loss function to account for the class imbalance. We have shown that this class balancing loss function leads to computational instability and a high false positive rate. As one of the methodological contributions, we have proposed a modified version of the class balancing loss function which is

computationally stable and help fix the problem of a high false positive rate in cases of extreme class imbalance.

UNAVAILABILITY OF ACCURATELY ANNOTATED TRAINING DATA the success of deep learning in natural image analysis is partly due to the existence of publicly available datasets. Deep learning networks learn from experience and require a lot of data for the learning process to be successful. Medical data comes with two main challenges, that is due to privacy policies associated with medical information there are very few publicly available medical datasets which makes it difficult to compare results in different experiments from different researchers. In addition, medical datasets are domain-specific and require a certain level of expertise to be able to annotate them. This makes obtaining accurately annotated training data very expensive and time-consuming. Annotating the vascular network, its centerlines, and bifurcation points are extremely slow and sometimes even not possible from a technical perspective. Drawing inspiration from the work of Schneider et al. [41] we have generated synthetic data with annotations for the vascular tree, its centerlines, and bifurcation points as part of this work. We have also shown that by using the synthetic data in a pre-training step we can use achieve high segmentation accuracy with little to no manually annotated data. Follow-up research by Todorov et al. [52] in Appendix A.3 has also shown the success of the synthetic data in a transfer learning experiment where deep networks are trained on the synthetic data and tested on mouse brain data. We have made the generated data publicly available and serve as the first publicly available 3-D vessel data with accurate annotations.

In our final set of experiments presented in Chapter 5 we discuss the crucial role of collateral flow in sustaining blood flow in the ischaemic areas in acute, subacute, or chronic phases after an ischaemic stroke or transient ischaemic attack. We highlighted the need for clinicians to determine the quality of the collateral circulation due to its role as a key factor in determining the likelihood of a successful reperfusion, favorable clinical outcome, and infarct growth in the early time window and impact the chosen stroke care model - that is the decision to transport or treat eligible patients immediately. We identified time consumption and inconsistent grading as the main challenges associated with the current clinical practice of grading collateral flow in stroke patients through manual inspection. We have provided

among other solutions a two-step approach toward automating the collateral flow grading task. The first step involves region of interest (ROI) extraction and a follow-up step is used to classify the extracted ROI as one of three gradings (poor = 0, medium = 1, good = 2). We have proposed a deep reinforcement learning approach for the ROI extraction task and have shown promising results in this task. For the classification of the extracted ROI, we have presented experiments using different configurations of feature extraction strategies and classifiers. We also examined a cascaded approach where we merged poor and medium (0, and 1) flow classes in a preliminary step and a follow-up step of separating the two classes. Our results show that using the extracted ROI improved the classification accuracy (from 55% to 72%) when compared to using the full image. The cascaded approach also showed a slight advantage over direct three-class prediction (72% vs 70%).

7

CONCLUSION AND OUTLOOK

This work covers medical image analysis topics related to brain vascular network analysis. The vascular network is a complex system and the fine nature of vessels makes the analysis in this domain very complicated. Intending to push the state of the art in vascular analysis, the methodological improvements proposed in this work are not to reinvent the wheel but aimed at addressing major challenges faced when using existing state-of-the-art deep learning architectures in vascular analysis.

Regarding the problem of computation time and memory consumption, the proposed methodological improvements have already proven very successful. We can however gain more improvements by utilizing a cross-hair filter which does not take into account all the information on the orthogonal planes but rather samples from these planes. This will lead to a loss of information and will require further experiments to measure the impact on segmentation or classification accuracy. Still, on the methodological side, the proposed arterial generative model in Schneider et al. [41], though works well, takes a long time and requires a post-processing step to convert the graph information to image information. Deep learning generative models like generative adversarial networks (GANs) and variational autoencoders (VAEs) that learn from existing clinical data could help improve the variation and quality of the synthetically generated dataset. Further experiments in this direction will help ascertain the suitability of these network architectures as models for generating synthetic vascular data.

Our work on collateral flow grading was limited mainly by the amount of data used in the study. Though the work covered many aspects methodologically, there are still other approaches that can be explored with improved data size. For example, the choice of reinforcement learning for the extraction of the region of interest is highly influenced by the size of the data used in the experiment. Supervised learning methods could lead to better ROI prediction accuracy but require more data during training. The results showed that CNN as a classifier performs better than

other classifiers. The CNN used is however very shallow due to the same data size limitation and a deeper CNN with more layers could further improve the classification accuracy. Another area that would need improvement is the data annotation. The interobserver agreement in the collateral flow grading needs to be thoroughly examined through a multi-center study to obtain better and standardized ground truth for training and proper evaluation of the methodologies. The work on collateral flow grading in this dissertation is the first attempt at fully automating the process and will serve as the basis for further research in this direction.

The goal of medical image analysis in general is to improve healthcare delivery and this goal can only be achieved when research findings are integrated into clinical routine. Besides the regulatory requirements, one major bottleneck in moving research into clinical practice is the inability to thoroughly test image processing methodologies in a clinical environment without interfering with the daily routine of clinicians. There are currently well-established libraries in the research field which makes it easy to implement ideas within a short time. These libraries have similar APIs which makes it easy to switch from one to the other with a very short learning curve. An interesting direction for future work will be to study the clinical environment and investigate ways in which clinical trials in medical image processing can be fully automated without interfering with the existing work routine. A possible outcome of such research will be a standardized translational tool or pipeline that would integrate easily into existing clinical systems and would provide an environment for easy clinical trials in the field.

Part IV

APPENDIX

A

PEER-REVIEWED CO-AUTHOR JOURNAL AND CONFERENCE PUBLICATIONS

A.1 TRANSFER LEARNING FROM SYNTHETIC DATA REDUCES NEED FOR LABELS TO SEGMENT BRAIN VASCULATURE AND NEURAL PATHWAYS IN 3D

Transfer learning from synthetic data reduces need for labels to segment brain vasculature and neural pathways in 3D

Johannes C. Paetzold[*1,†], **Oliver Schoppe**[*1], **Rami Al-Maskari**[1], **Giles Tetteh**[1], **Velizar Efremov**[1], **Mihail I. Todorov**[2], **Ruiyao Cai**[2], **Hongcheng Mai**[2], **Zhouyi Rong**[2], **Ali Ertuerk**[2], **Bjoern H. Menze**[1,†]

[1] *TranslaTUM and Department of Computer Science, Technical University of Munich, Germany*

[2] *Institute for Stroke and Dementia Research, Ludwig Maximilian University of Munich, Germany*

[†] *Correspondence to johannes.paetzold@tum.de and bjoern.menze@tum.de*

Editors: Under Review for MIDL 2019

Abstract

Novel microscopic techniques yield high-resolution volumetric scans of complex anatomical structures such as the blood vasculature or the nervous system. Here, we show how transfer learning and synthetic data generation can be used to train deep neural networks to segment these structures successfully in the absence of or with very limited training data.

Keywords: Deep learning, transfer learning, synthetic data, vasculature, neural pathways.

1. Introduction

Recent advances in tissue-clearing (Ertürk et al., 2012; Chung and Deisseroth, 2013) combined with 3D light-sheet microscopy (*3D LSM*) overcome previous imaging limitations: they enable volumetric acquisition at cellular resolution of entire organisms (Cai et al., 2018; Pan et al., 2019; Stefaniuk et al., 2016; Mano et al., 2018). This yields unprecedented insight into the micro-anatomy at the macro-scale, e.g., to study highly connected structures like the brain vasculature or the peripheral nervous system. Differences in these structures have been associated with a wide range of disorders (Joutel et al., 2010; Li et al., 2010). Thus, segmentation and characterization of these anatomical structures is crucial to study causes and effects of such pathologies. However, manual segmentation of complex structures is very time-consuming, especially in high-resolution volumetric scans. While this motivates the need for deep learning it also implies a high cost of labeling. Here, we substantially reduce the need for manually labeled training data using transfer learning, an approach gaining attention (Van Opbroek et al., 2015; Khan et al., 2019). In short, we show that training deep networks on synthetic data is already sufficient to learn the basic underlying task across different anatomical structures, species, and imaging modalities.

2. Methods

Here, we present results from three widely different applications: human brain vessels (MRI), mouse brain vessels and the mouse peripheral nervous system (both *3D LSM*).

[*] Contributed equally

The same network was trained either on a small labeled set from the respective application ("real data"), on synthetically generated data, or on a combination of both. The synthetic data used is identical for all three applications. We chose DeepVesselNet as our architecture; the schedule for pre-training on synthetic data and refinement on real data match the methods of (Tetteh et al., 2018). The methods for generation of synthetic training data is described in (Schneider et al., 2012). MRI scans from human brain vasculature are taken from (Tetteh et al., 2018) (voxel size: 300μm x 300μm x 600μm). Volumetric scans of the brain vasculature (voxel size: $(3\mu\text{m})^3$) and the peripheral nervous system (voxel size: $(10\mu\text{m})^3$) were obtained using DISCO tissue clearing and fluorescent light-sheet microscopy as described in (Cai et al., 2018). Representative 2D cross-sections of the synthetic data and segmentations of all three applications are shown in Figure 1.

3. Results

Transfer learning from synthetic data (Table 1, Part 1). For segmenting the human vasculature from MRI scans, training the net on the synthetic data alone yields very good results, 81% in F1-score (note: the synthetic data set had been designed for this application). Training on the real data for this application yields a higher F1-score of 86%. The best result (87%), however, is achieved by a combination of both: pre-training on synthetic data and fine-tuning on real data. Interestingly, the network also converges about 50% faster in this case (data not shown). Motivated by this observation, we repeated this experiment for *3D LSM* scans of the mouse brain vasculature. Again, the same pattern can be observed and the combination of synthetic with real data (F1-score of 76%) outperforms synthetic data (71%) or real data alone (73%). Taking the approach yet further, we applied the approach to *3D LSM* full body scans of the peripheral nervous system of a mouse. While training on synthetic data alone was not very successful (16%) as compared to real data (49%), the gain from combining both was almost completely additive (64%).

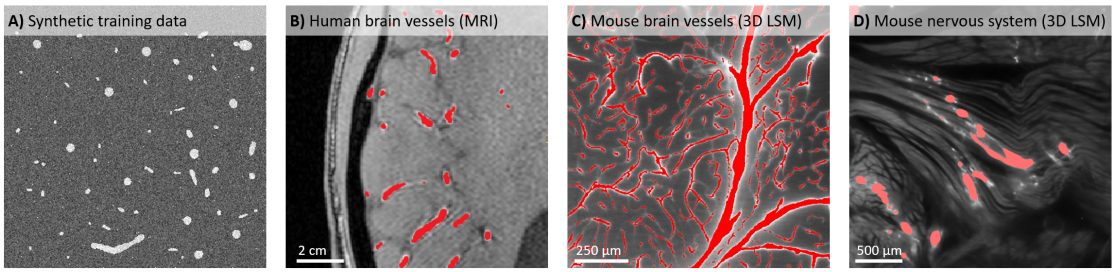


Figure 1: A) Synthetic training data was designed to resemble vasculature of human brain in MRI scans. B-D) Predicted segmentations of 3 different applications: MRI scans of human brain vasculature (B), *3D LSM* of mouse brain vasculature (C), and peripheral nervous system (D; shown here: innervated muscle fibres)

Transfer learning across domains (Table 1, Part 2). Here, we trained the network on a combination of synthetic data and the real data from a given application and then predicted on data from another application. When predicting on human vasculatures from

MRI scans, the refinement step on real data from another application after pre-training on synthetic data leads to worse results (left column: 43% and 36%) compared to training on synthetic data alone (81%, see Part 1). However, when training the model on synthetic data and real data of human vessels in MRI scans (first row of Part 2), the performance on *3D LSM* scans of mouse brain vessels (72%) or the mouse peripheral nervous system (49%) is about as good as when trained on the respective real data alone. Also, while the domain transfer from mouse vasculature to mouse nervous system only yields mediocre results (35%), it works well the other way around: refining a model trained on synthetic data with real data from the nervous system to segment brain vessels almost works as well (75%) as if it had been refined on data within the same domain (76%, see Part 1).

	Training set	**Application #1** Human brain Vasculature MRI	**Application #2** Mouse brain Vasculature 3D microscopy	**Application #3** Mouse body Neural pathways 3D microscopy
Part 1) Tranfer learning from synthetic data within domain	Synthetic data only	81%	71%	16%
	Real data only	86%	73%	49%
	Synthetic + real data	**87%**	**76%**	**64%**
Part 2) Transfer learning across application domains	Synthetic + human vessel MRI data	*n/a*	72%	49%
	Synthetic + mouse vessel microscopy data	43%	*n/a*	35%
	Synthetic + mouse neuron microscopy data	36%	75%	*n/a*

Table 1: Quality of predicted segmentations (F1-score) for 3 different applications

4. Discussion

Our results demonstrate how pre-training on synthetically generated data can accelerate model convergence and boost the overall segmentation performance. For a given desired performance, this thus means a reduced need for manually labeled training data, which is very expensive for complex structures in 3D scans. Importantly, a single synthetic data set that was originally designed to represent human vessels also works well for applications from different species, anatomical structures, and imaging modalities. This suggests that the features learned from the synthetic data are of general use for the abstract segmentation tasks, highlighting the generalizability of the approach. Thus, the expensively labeled data for a given application does not have to be used to learn a basic task but rather can be preserved for refining the pre-trained model to the specifics of the application (such as contrast, noise, background structures). Interestingly, this approach may also be of use in cases where no training data is available at all. For instance, we could show that a model trained on synthetic data and real data from another application can match the performance of a model trained from scratch on real data from the application of interest. Together, these results highlight the importance of transfer learning towards the goal of resolving a key bottleneck in adoption of deep learning: the high cost of data annotation.

Acknowledgments

This work was supported the German Federal Ministry of Education and Research via the Software Campus initiative (to O.S.), the Vascular Dementia Research Foundation, Synergy Excellence Cluster Munich (SyNergy), ERA-Net Neuron (01EW1501A to A.E.). Furthermore, NVIDIA supported this work with a Titan XP GPU via the GPU Grant Program.

References

Ruiyao Cai, Chenchen Pan, Alireza Ghasemigharagoz, Mihail Ivilinov Todorov, Benjamin Förstera, Shan Zhao, Harsharan S Bhatia, Arnaldo Parra-Damas, Leander Mrowka, Delphine Theodorou, et al. Panoptic imaging of transparent mice reveals whole-body neuronal projections and skull–meninges connections. Technical report, Nature Publishing Group, 2018.

Kwanghun Chung and Karl Deisseroth. Clarity for mapping the nervous system. *Nature methods*, 10(6):508, 2013.

Ali Ertürk, Klaus Becker, Nina Jährling, Christoph P Mauch, Caroline D Hojer, Jackson G Egen, Farida Hellal, Frank Bradke, Morgan Sheng, and Hans-Ulrich Dodt. Three-dimensional imaging of solvent-cleared organs using 3disco. *Nature protocols*, 7(11):1983, 2012.

Anne Joutel, Marie Monet-Leprêtre, Claudia Gosele, Céline Baron-Menguy, Annette Hammes, Sabine Schmidt, Barbara Lemaire-Carrette, Valérie Domenga, Andreas Schedl, Pierre Lacombe, et al. Cerebrovascular dysfunction and microcirculation rarefaction precede white matter lesions in a mouse genetic model of cerebral ischemic small vessel disease. *The Journal of clinical investigation*, 120(2):433–445, 2010.

SanaUllah Khan, Naveed Islam, Zahoor Jan, Ikram Ud Din, and Joel JP C Rodrigues. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. *Pattern Recognition Letters*, 2019.

Weiguo Li, Roshini Prakash, Aisha I Kelly-Cobbs, Safia Ogbi, Anna Kozak, Azza B El-Remessy, Derek A Schreihofer, Susan C Fagan, and Adviye Ergul. Adaptive cerebral neovascularization in a model of type 2 diabetes: relevance to focal cerebral ischemia. *Diabetes*, 59(1):228–235, 2010.

Tomoyuki Mano, Alexandre Albanese, Hans-Ulrich Dodt, Ali Erturk, Viviana Gradinaru, Jennifer B Treweek, Atsushi Miyawaki, Kwanghun Chung, and Hiroki R Ueda. Whole-brain analysis of cells and circuits by tissue clearing and light-sheet microscopy. *Journal of Neuroscience*, 38(44):9330–9337, 2018.

Chenchen Pan, Oliver Schoppe, Arnaldo Parra-Damas, Ruiyao Cai, Mihail Ivilinov Todorov, Gabor Gondi, Bettina von Neubeck, Alireza Ghasemi, Madita Alice Reimer, Javier Coronel, et al. Deep learning reveals cancer metastasis and therapeutic antibody targeting in whole body. *bioRxiv*, page 541862, 2019.

Matthias Schneider, Johannes Reichold, Bruno Weber, Gábor Székely, and Sven Hirsch. Tissue metabolism driven arterial tree generation. *Medical image analysis*, 16(7):1397–1414, 2012.

Marzena Stefaniuk, Emilio J Gualda, Monika Pawlowska, Diana Legutko, Paweł Matryba, Paulina Koza, Witold Konopka, Dorota Owczarek, Marcin Wawrzyniak, Pablo Loza-Alvarez, et al. Light-sheet microscopy imaging of a whole cleared rat brain with thy1-gfp transgene. *Scientific reports*, 6:28209, 2016.

Giles Tetteh, Velizar Efremov, Nils D Forkert, Matthias Schneider, Jan Kirschke, Bruno Weber, Claus Zimmer, Marie Piraud, and Bjoern H Menze. Deepvesselnet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. *arXiv preprint arXiv:1803.09340*, 2018.

Annegreet Van Opbroek, M Arfan Ikram, Meike W Vernooij, and Marleen De Bruijne. Transfer learning improves supervised image segmentation across imaging protocols. *IEEE transactions on medical imaging*, 34(5):1018–1030, 2015.

A.2 CLDICE - A NOVEL CONNECTIVITY-PRESERVING LOSS FUNCTION FOR VESSEL SEGMENTATION

clDice - a Novel Connectivity-Preserving Loss Function for Vessel Segmentation

Johannes C. Paetzold[*]
Technical University Munich
johannes.paetzold@tum.de

Suprosanna Shit[*†]
Technical University Munich
suprosanna.shit@tum.de

Ivan Ezhov
Technical University Munich
ivan.ezhov@tum.de

Giles Tetteh
Technical University Munich
giles.tetteh@tum.de

Ali Ertürk
Helmholtz Zentrum Munich
erturk@helmholtz-muenchen.de

Bjoern Menze
Technical University Munich
bjoern.menze@tum.de

Abstract

Accurate segmentation of vascular structures is an emerging research topic with relevance to clinical and biological research. The connectedness of the segmented vessels is often the most significant property for many applications such as disease modeling for neurodegeneration and stroke. We introduce a novel metric namely *clDice*, which is calculated on the intersection of centerlines and volumes as opposed to the traditional dice, which is calculated on volumes only. Firstly, we tested state-of-the-art vessel segmentation networks using the proposed metric as evaluation criteria and show that it captures vascular network properties superior to traditional metrics, such as the dice-coefficient. Secondly, we propose a differentiable form of *clDice* as a loss function for vessel segmentation. We find that training on *clDice* leads to segmentation with more accurate connectivity information, higher graph similarity and often superior volumetric scores.

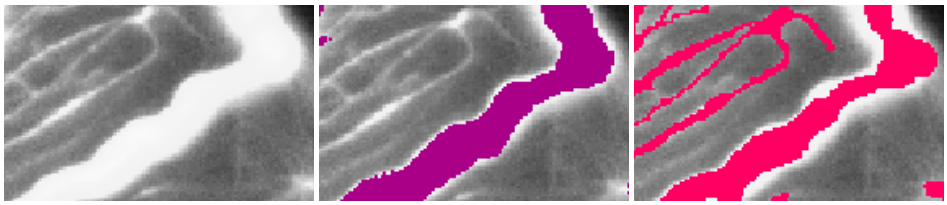
1 Introduction

Segmentation of blood vessels is a key step in many clinical and biological applications such as analyzing neurodegenerative diseases, e.g. Alzheimer's disease [1], brain-vessel, and stroke modeling [2]. The two most commonly used categories of quantitative performance measures are a) overlap based distance measures such as dice-score, precision, recall, and Jaccard index; and b) volumetric distance measures such as the Hausdorff distance and the Mahalanobis distance [3, 4, 5, 6, 7].

However, in many vessel segmentation applications, the most important properties are the connectivity of the vascular network segments. Traditional scores, e.g. dice and Jaccard rely on the average voxel-wise hit or miss prediction [8]. On the other hand, in a task like vascular network extraction, a proper sequence of hits in the voxel domain is preferred over spurious hits. Further, a globally averaged metric does not equally weight vessels with large, medium and small radii. In real datasets, where vessels of wide radius ranges exist, e.g. 30 μm for arterioles [6, 9] and 5 μm for capillaries, training on a globally averaged loss induces a strong bias towards the volumetric segmentation of

[*]Authors contributed equally and should be considered as joint first authors.
[†]Corresponding Author.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.



(a) (b) (c)

Figure 1: **Motivation**, (a) Shows an examplary 2D slice of real microscopic data, (b) and (c) are two random segmentation results which achieve similar dice scores (not from our presented model). Note that (b) does not capture any of the small vessels while segmenting the large vessel very accurately, on the other side segmentation (c) captures all vessels in the image while being a less accurate on the diameter of the large vessel.

large vessels. This is pronounced in imaging modalities like fluorescence microscopy, where the image intensities of arteriole surrounding tissue are higher than the intensity within capillaries [6]. In Figure 1, an example illustrates the suboptimality of traditional scores in some scenarios.

Furthermore, the most traditional metrics are ambiguous when some of the objects of interest are of the same order as the resolution of the signal. Single-voxel shifts in prediction change the local metric score significantly, thus making the metric difficult to interpret [8]. In this context of a vascular network extraction task, we ask the following research questions:

Q1. What is a good connectivity-aware metric to benchmark vessel-segmentation algorithms?

Q2. How can we leverage this metric in a loss function to improve connectivity in vessel-segmentation?

2 Methods

In this section we first introduce the *clDice* as a metric and subsequently introduce a differentiable loss function namely *soft-clDice*.

***clDice* Metric :** We propose a novel connectivity-preserving metric to evaluate vessel segmentation, based on intersecting centerlines of vessels with vessel volumes. We call this metric a centerline-in-volume-dice-coefficient or *clDice* in short. We consider two binary volumes: first, the ground truth label map (L), and second, the predicted segmentation volume (P). The centerlines cl_P and cl_L are extracted from P and L respectively. Subsequently, we compute the fraction of cl_L that lies within P, which we call $cl_L 2vol_P$ and vice-a-versa to get $cl_P 2vol_L$ [c.f. Algorithm 1]. We observe that $cl_P 2vol_L$ is very susceptible to false positives in the prediction while $cl_L 2vol_P$ is susceptible to false negatives. Therefore, we proceed to interpret $cl_L 2vol_P$ as precision and $cl_P 2vol_L$ as recall. Since we want to maximize both precision and recall, we formulate it symmetrically similar to the dice coefficient in Equation 1. This leads us to the final expression of *clDice* in Equation 2.

$$Dice = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \qquad (1) \qquad clDice = 2 \times \frac{cl_P 2vol_L \times cl_L 2vol_P}{cl_P 2vol_L + cl_L 2vol_P} \qquad (2)$$

***soft-clDice* Loss :** The centerline can be extracted through Euclidean distance transform or via repeated morphological thinning. Although Euclidean distance transform has been used in multiple occassion [10] to induce skeletons, it is a discrete operation and an end-to-end differentiable approximation remains unsolved, which prevents us from using it in our loss function. On the contrary, morphological thinning consists of dilation and erosion operations. Further, min- and max filters are commonly used as the greyscale alternative of morphological dilation and erosion. Motivated by this fact we replace dilation and erosion operations with iterative min- and max-pooling. This allows us to leverage *clDice* to extract a parameter-free, morphologically motivated soft-centerline on greyscale valued data. We call this loss *soft-clDice* and describe it in Algorithm 1 and 2. We determine the hyper-parameter k to be in the range of the maximum radius for

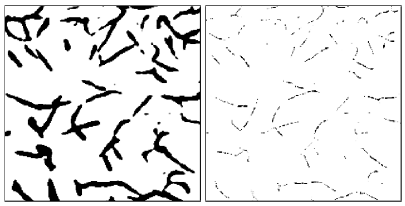


Figure 2: *2D slice of the soft-centerline*(right) of a real valued class probability map (left).

the vessel like structure. In our experiment it is 5 for the synthetic and real data. Choosing a larger k does not reduce performance but increases computation time, on the other hand a too low k leads to incomplete skeletonization.

Algorithm 1: *soft-clDice*

Input : P, L

$cl_P \leftarrow$ *soft-centerline*(P)

$cl_L \leftarrow$ *soft-centerline*(L)

$cl_P 2vol_L \leftarrow \frac{|cl_P \circ L|+\epsilon}{|cl_P|+\epsilon}$

$cl_L 2vol_P \leftarrow \frac{|cl_L \circ P|+\epsilon}{|cl_L|+\epsilon}$

$$clDice \leftarrow 2 \times \frac{cl_P 2vol_L \times cl_L 2vol_P}{cl_P 2vol_L + cl_L 2vol_P}$$

Output : $clDice$

Algorithm 2: *soft-centerline*

Input : I, k

$I' \leftarrow maxpool(minpool(I))$

$cl \leftarrow ReLU(I - I')$

for $i \leftarrow 0$ **to** k **do**

  $I \leftarrow minpool(I)$

  $I' \leftarrow maxpool(minpool(I))$

  $cl \leftarrow cl + cl \circ ReLU(I - I')$

end

Output : cl

Vessel Segmentation: We evaluate the proposed *clDice* metric on two state-of-the-art 3D veseel segmentation networks i) a 3D U-net[11], and ii) a 3D fully connected network (FCN)[12]. We used generalized *soft-Dice* [13, 14] to train our baseline model for the vessel segmentation. Since our objective here is to achieve accurate segmentation, while giving vascular connection more importance, we add our proposed *soft-clDice* with *soft-Dice* as following

$$\mathcal{L}_c = 0.5(\textit{soft-Dice} + \textit{soft-clDice}) \tag{3}$$

In stark contrast to previous works, where vessel segmentation and centerline prediction has been learned jointly as a multi-task learning [15], or which only learned vessel-centerlines (or trees [12]), we are not interested in learning the centerlines. We are interested in learning a whole vessel segmentation, where the connections between individual vessels are robust and complete.

3 Experiments

Dataset: We test our proposed metric and loss function on a synthetic and a real dataset. The generation of the synthetic data is described in [16], additionally, we add a Gaussian noise term to this generated data. The real dataset consists of multi-channel volumetric scans of the brain vasculature (voxel size: $(3\mu m^3)$, which were obtained using light-sheet microscopy of tissue cleared Murine brains, as introduced in [17]. We performed experiments on a synthetic dataset using fifteen single-channel volumes for training, two for validation and five for testing, each of the size $325 \times 304 \times 600$ pixels. On the real data we used both single and two-channel inputs, the inputs correspond to different fluorescent stains, which have been shown to contain complimentary information [6]. Eleven volumes were used for training, two for validation and four for testing, each of the size $500 \times 500 \times 50$ pixels.

Evaluation Metric: We report overlap based metrics such as the Dice coefficient, Jaccard index(IOU) and Accuracy along with our proposed *clDice* for all the experiment settings. Additionally, we extract a vascular graph from the centerline of the predicted segmentation and compute relative accuracy of total vascular network length (Dist.), the number of detected bifurcation points (Bifurc.) and endpoints (End Pt.) compared to the ground truth.

Results & Discussion: We trained a Unet and a FCN in different scenarios of identical settings and datasets. From Table 1 we observe that the inclusion of *soft-clDice* loss not only leads to a higher *clDice* in all cases, but also performs better than the standalone *soft-dice* in terms of dice coefficient and IOU. We also observe that *soft-clDice* improves the extracted network properties significantly for real data. We do not see any systematic change in synthetic data after adding *soft-clDice*. We attribute this to the fact that the synthetic data has higher signal-to-noise ratio and lacks significant illumination variation.

4 Conclusion

This abstract introduces a novel connectivity-preserving metric *clDice* for vessel segmentation. We use the new metric to evaluate segmentation quality, and in a loss function, to train state-of-the-art networks on real and synthetic data. We find that training on *soft-clDice* leads to vessel segmentation

Table 1: Experimental results for 3D U-nets and 3D FCNs on synthetic and real data. We observe a consistent performance improvement for real data with the combination of *soft-clDice* and *soft-dice*.

Data	Loss	Network	Dice	clDice	IOU	Acc.	Dist.	Bifurc.	End Pt.
Synthetic	*soft-dice*	FCN, 1 ch	**99.41**	99.45	**98.83**	99.97	0.92	0.91	0.91
		Unet, 1 ch	**99.61**	99.90	97.23	**99.98**	0.88	0.86	0.89
	$\mathcal{L}_c$	FCN, 1 ch	99.16	**99.77**	98.34	99.96	0.92	0.91	0.92
		Unet, 1 ch	98.73	99.90	**97.49**	99.94	0.88	0.86	0.88
Real data	*soft-dice*	FCN, 1 ch	75.28	90.98	60.35	89.88	0.87	0.72	0.81
		FCN, 2 ch	78.54	92.03	64.67	91.66	0.90	0.82	0.84
		Unet, 1 ch	**87.11**	95.03	**77.17**	95.78	0.92	0.82	0.97
		Unet, 2 ch	80.20	93.05	66.94	92.33	0.95	**0.93**	0.70
	$\mathcal{L}_c$	FCN, 1 ch	**85.57**	**96.16**	**74.78**	**95.09**	**0.97**	**0.88**	**0.97**
		FCN, 2 ch	**85.28**	**95.75**	**74.34**	**94.91**	**0.91**	**0.91**	**0.97**
		Unet, 1 ch	86.94	**95.28**	76.89	**95.86**	**0.94**	**0.83**	0.97
		Unet, 2 ch	**83.96**	**96.10**	**72.36**	**94.18**	**0.96**	0.89	**0.85**

with more accurate connectivity information, higher graph similarity and similar to better volumetric scores. More importantly *clDice* and *soft-clDice* can be readily deployed in other tree-structured object segmentation tasks such as neuron segmentation and bronchial tract segmentation.

Acknowledgments

Suprosanna Shit and Ivan Ezhov are supported by the Translational Brain Imaging Training Network (TRABIT) under the European Union's 'Horizon 2020' research & innovation program (Grant agreement ID: 765148). The authors would like to thank the Deutsche Forschungsgemeinschaft (DFG; ME3511/3-1). We thank Mihail I. Todorov, Amirhossein Bayat and Oliver Schoppe.

References

[1] Jesse M Hunter et al. Morphological and pathological evolution of the brain microcirculation in aging and Alzheimer's disease. *PloS one*, 7(5):e36893, 2012.

[2] Anne Joutel et al. Cerebrovascular dysfunction and microcirculation rarefaction precede white matter lesions in a mouse genetic model of cerebral ischemic small vessel disease. *The Journal of Clinical Investigation*, 120(2):433–445, 2010.

[3] Cemil Kirbas and Francis Quek. A review of vessel extraction techniques and algorithms. *ACM Computing Surveys (CSUR)*, 36(2):81–121, 2004.

[4] Matthias Schneider et al. Joint 3-D vessel segmentation and centerline extraction using oblique Hough forests with steerable filters. *Medical Image Analysis*, 19(1):220–249, 2015.

[5] Renzo Phellan et al. Vascular segmentation in tof mra images of the brain using a deep convolutional neural network. In *MICCAI Workshop*, pages 39–46. Springer, 2017.

[6] Mihail Ivilinov Todorov et al. Automated analysis of whole brain vasculature using machine learning. *bioRxiv*, page 613257, 2019.

[7] Kai Hu et al. Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function. *Neurocomputing*, 309:179–191, 2018.

[8] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 15(1):29, 2015.

[9] Antonino Paolo Di Giovanna et al. Whole-brain vasculature reconstruction at the single capillary level. *Scientific reports*, 8(1):12573, 2018.

[10] Fernando Navarro et al. Shape-aware complementary-task learning for multi-organ segmentation. In *International Workshop on MLMI*, pages 620–627. Springer, 2019.

[11] Özgün Çiçek and Aothers. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *MICCAI*, pages 424–432. Springer, 2016.

[12] Giles Tetteh et al. Deepvesselnet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. *arXiv preprint arXiv:1803.09340*, 2018.

[13] Fausto Milletari et al. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, pages 565–571. IEEE, 2016.

[14] Carole H Sudre et al. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *MICCAI Workshop*, pages 240–248. Springer, 2017.

[15] Fatmatülzehra Uslu and Anil Anthony Bharath. A multi-task network to detect junctions in retinal vasculature. In *MICCAI*, pages 92–100. Springer, 2018.

[16] Matthias Schneider et al. Tissue metabolism driven arterial tree generation. *Medical Image Analysis*, 16(7):1397–1414, 2012.

[17] Ali Ertürk et al. Three-dimensional imaging of solvent-cleared organs using 3DISCO. *Nature protocols*, 7(11):1983, 2012.

A.3 MACHINE LEARNING ANALYSIS OF WHOLE MOUSE BRAIN VASCULATURE

nature methods

ARTICLES

https://doi.org/10.1038/s41592-020-0792-1

Check for updates

Machine learning analysis of whole mouse brain vasculature

Mihail Ivilinov Todorov[1,2,3,10], Johannes Christian Paetzold[4,5,6,10], Oliver Schoppe[4,5], Giles Tetteh[4], Suprosanna Shit[4,5,6], Velizar Efremov[4,7], Katalin Todorov-Völgyi[2], Marco Düring[2,8], Martin Dichgans[2,8,9], Marie Piraud[4], Bjoern Menze[4,5,6,11] ✉ and Ali Ertürk[1,2,8,11] ✉

Tissue clearing methods enable the imaging of biological specimens without sectioning. However, reliable and scalable analysis of large imaging datasets in three dimensions remains a challenge. Here we developed a deep learning-based framework to quantify and analyze brain vasculature, named Vessel Segmentation & Analysis Pipeline (VesSAP). Our pipeline uses a convolutional neural network (CNN) with a transfer learning approach for segmentation and achieves human-level accuracy. By using VesSAP, we analyzed the vascular features of whole C57BL/6J, CD1 and BALB/c mouse brains at the micrometer scale after registering them to the Allen mouse brain atlas. We report evidence of secondary intracranial collateral vascularization in CD1 mice and find reduced vascularization of the brainstem in comparison to the cerebrum. Thus, VesSAP enables unbiased and scalable quantifications of the angioarchitecture of cleared mouse brains and yields biological insights into the vascular function of the brain.

Changes in cerebrovascular structures are key indicators for a large number of diseases affecting the brain. Primary angiopathies, vascular risk factors (for example, diabetes), traumatic brain injury, vascular occlusion and stroke all affect the function of the brain's vascular network[1–3]. The hallmarks of Alzheimer's disease, including tauopathy and amyloidopathy, can also lead to aberrant remodeling of blood vessels[1,4], allowing capillary rarefaction to be used as a marker for vascular damages[5]. Therefore, quantitative analysis of the entire brain vasculature is pivotal to developing a better understanding of brain function in physiological and pathological states. However, quantifying micrometer-scale changes in the cerebrovascular network of the brain has been difficult for two main reasons.

First, labeling and imaging of the complete mouse brain vasculature down to the smallest blood vessels has not yet been achieved. Magnetic resonance imaging (MRI), micro-computed tomography (micro-CT) and optical coherence tomography do not have sufficient resolution to capture capillaries in bulk tissue[6–8]. Fluorescent microscopy provides higher resolution, but can typically only be applied to tissue sections up to 200 µm in thickness[9]. Recent advances in tissue clearing could overcome this problem[10], but so far there has been no systematic description of all vessels of all sizes in an entire brain in three dimensions (3D).

The second challenge relates to the automated analysis of large 3D imaging datasets with substantial variance in signal intensity and signal-to-noise ratio (SNR) at different depths. Simple intensity- and shape-based filtering approaches such as Frangi's vesselness filters and more advanced image processing methods with local spatial adaptation cannot reliably differentiate vessels from background in whole-brain scans[11,12]. Finally, imaging of the complete vascular network of the brain at capillary resolution results in datasets of terabyte size. Established image processing methods do not scale well to terabyte-sized image volumes, as they do not generalize well to large images, and require intensive manual fine-tuning[13–15].

Here we present VesSAP (Vessel Segmentation & Analysis Pipeline), a deep learning-based method for automated analysis of the entire mouse brain vasculature, overcoming the above limitations. VesSAP encompasses three major steps: (1) staining, clearing and imaging of the mouse brain vasculature down to the capillary level with two different dyes: wheat germ agglutinin (WGA) and Evans blue (EB); (2) automatic segmentation and tracing of the whole-brain vasculature data via CNNs; and (3) extraction of vascular features for hundreds of brain regions after registration of the data to the Allen brain atlas (Fig. 1). Our deep learning-based approach for network extraction in cleared tissue is robust, despite variations in signal intensities and structures, outperforms previous filter-based methods and reaches the quality of segmentation achieved by human annotators. We applied VesSAP to the three commonly used mouse strains C57BL/6J, CD1 and BALB/c.

Results

Vascular staining, DISCO clearing and imaging. To reliably stain the entire vasculature, we used WGA and EB dyes, which can be visualized in different fluorescence channels. We injected EB dye into live mice 12 h before WGA perfusion, allowing its long-term circulation to mark vessels under physiological conditions[16], while we perfused mice with WGA during fixation. We then performed

[1]Institute for Tissue Engineering and Regenerative Medicine (iTERM), Helmholtz Zentrum München, Neuherberg, Germany. [2]Institute for Stroke and Dementia Research (ISD), Ludwig-Maximilians-Universität (LMU), Munich, Germany. [3]Graduate School of Neuroscience (GSN), Munich, Germany. [4]Department of Computer Science, Technical University of Munich (TUM), Munich, Germany. [5]Center for Translational Cancer Research of the TUM (TranslaTUM), Munich, Germany. [6]Munich School of Bioengineering, Technical University of Munich (TUM), Munich, Germany. [7]Institute of Pharmacology and Toxicology, University of Zurich (UZH), Zurich, Switzerland. [8]Munich Cluster for Systems Neurology (SyNergy), Munich, Germany. [9]German Center for Neurodegenerative Diseases (DZNE), Munich, Germany. [10]These authors contributed equally: Mihail Ivilinov Todorov, Johannes Christian Paetzold. [11]These authors jointly supervised this work: Bjoern Menze, Ali Ertürk. ✉e-mail: bjoern.menze@tum.de; erturk@helmholtz-muenchen.de

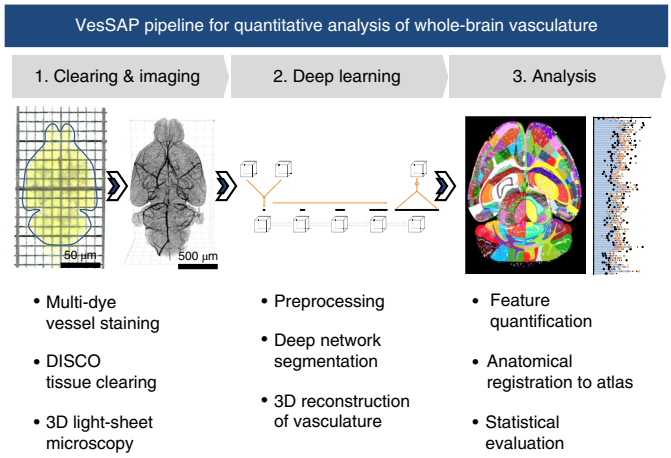


Fig. 1 | Summary of the VesSAP pipeline. The method consists of three modular steps: (1) multi-dye vessel staining and DISCO tissue clearing for high imaging quality using 3D light-sheet microscopy; (2) deep learning-based segmentation of blood vessels with 3D reconstruction; and (3) anatomical feature extraction and mapping of the entire vasculature to the Allen adult mouse brain atlas for statistical analysis.

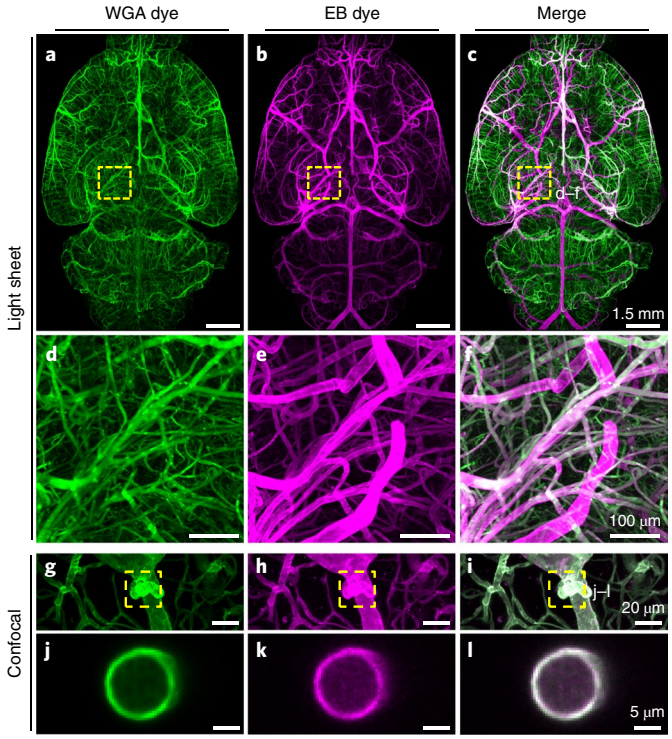


Fig. 2 | Enhancement of vascular staining using two complementary dyes. **a**–**c**, Maximum-intensity projections of automatically reconstructed tiling scans of WGA (**a**) and EB (**b**) signal in the same sample and the merged view (**c**). **d**–**f**, Magnified view of the boxed region in **c**. **g**–**l**, Confocal images of WGA- and EB-stained vessels and vascular wall (**g**–**i**, maximum-intensity projections at 112 µm; **j**–**l**, single slices of 1 µm corresponding to the boxed region in **i**). The experiment was performed on nine different mice with similar results.

3DISCO clearing[17] and light-sheet microscopy imaging of whole mouse brains (Fig. 2a–c and Supplementary Figs. 1 and 2). WGA highlighted microvessels, and EB predominantly stained major blood vessels, such as the middle cerebral artery and the circle of Willis (Fig. 2d–i and Supplementary Fig. 3). Merging the signals from the two dyes yielded more complete staining of the vasculature than relying on individual dyes alone (Fig. 2c,f and Supplementary Video 1). Staining with the two dyes was congruent in midsized vessels, with signals originating from the vessel wall layer (Fig. 2j–l and Supplementary Fig. 3a–c). When using WGA, we reached a higher SNR for microvessels than for bigger vessels. With EB, the SNR for small capillaries was lower but larger vessels reached a high SNR (Supplementary Fig. 4). Integrating the information from the two channels allowed acquisition of the entire vasculature and resulted in optimized SNR. We also compared the fluorescence signal quality of the WGA staining (targeting the complete endothelial glycocalyx lining[18]) to signal for a conventional vessel-specific antibody (anti-CD31, targeting endothelial cell–cell adhesion[19]) and found that WGA produced higher SNR for blood vessels in general (Supplementary Fig. 5).

Segmentation of volumetric images. To enable extraction of quantitative features of the vascular structure, vessels in acquired brain scans need to be segmented in 3D. Motivated by deep learning-based approaches in biomedical image data analysis[20–28], we used a five-layer CNN (Fig. 3a) to exploit the complementary signals of the two dyes to derive complete segmentation of the entire brain vasculature.

In the first step, the two input channels (WGA and EB) were concatenated. This yielded a matrix in which each voxel was characterized by two features. Then, each convolutional step integrated the information from a voxel's 3D neighborhood. We used full 3D convolutions[20] without further down- or upsampling and fewer trainable parameters than, for example, 3D U-Net and V-Net[29,30] to achieve high inference speeds. After the fourth convolution, the information from 50 features per voxel was combined with a convolutional layer with a kernel size of one and sigmoidal activation to estimate the likelihood that a given voxel represented a vessel. Subsequent binarization yielded the final segmentation. In both training and testing, the images were processed in subvolumes of $50 \times 100 \times 100$ pixels.

Deep neural networks often require large amounts of annotated data or many iterations of training. Here we circumvented this requirement with a transfer learning approach[31]. In short, we first pretrained the network on a large, synthetically generated vessel-like dataset (Supplementary Fig. 6)[32] and then refined it on a small number of manually annotated parts of real brain vessel scans. This approach reduced the training iterations on manually annotated training data.

To assess the quality of the segmentation, we compared the VesSAP CNN predictions to manually labeled ground truth and the predictions from alternative computational approaches (Table 1). We report voxel-wise segmentation metrics, namely, accuracy, F1 score[33], Jaccard coefficient and cl-F1, which weights the centerlines and volumes of the vessels (detailed in the Methods). In comparison to the ground truth, our network achieved an accuracy of 0.94 ± 0.01 and an F1 score of 0.84 ± 0.05 (for additional scores, see Table 1; all values are given as the mean $\pm$ s.d.). As controls, we implemented alternative state-of-the-art deep learning and classical methods. Our network outperformed classical Frangi filters[11] (accuracy, 0.85 ± 0.03; F1 score, 0.47 ± 0.18), as well as recent methods based on local spatial context via Markov random fields[13,34] (accuracy, 0.85 ± 0.03; F1 score, 0.48 ± 0.04). VesSAP achieved similar performance in comparison to 3D U-Net and V-Net architectures, which require substantially more trainable parameters (3D U-Net: accuracy, 0.95 ± 0.01; F1 score, 0.85 ± 0.03; V-Net: accuracy, 0.95 ± 0.02; F1 score, 0.86 ± 0.07; no statistical difference in comparison to the VesSAP CNN: two-sided t test, all $P > 0.3$). However, the VesSAP CNN substantially outperformed the other architectures in terms of speed, being ~20 and ~50 times faster in the feedforward path than

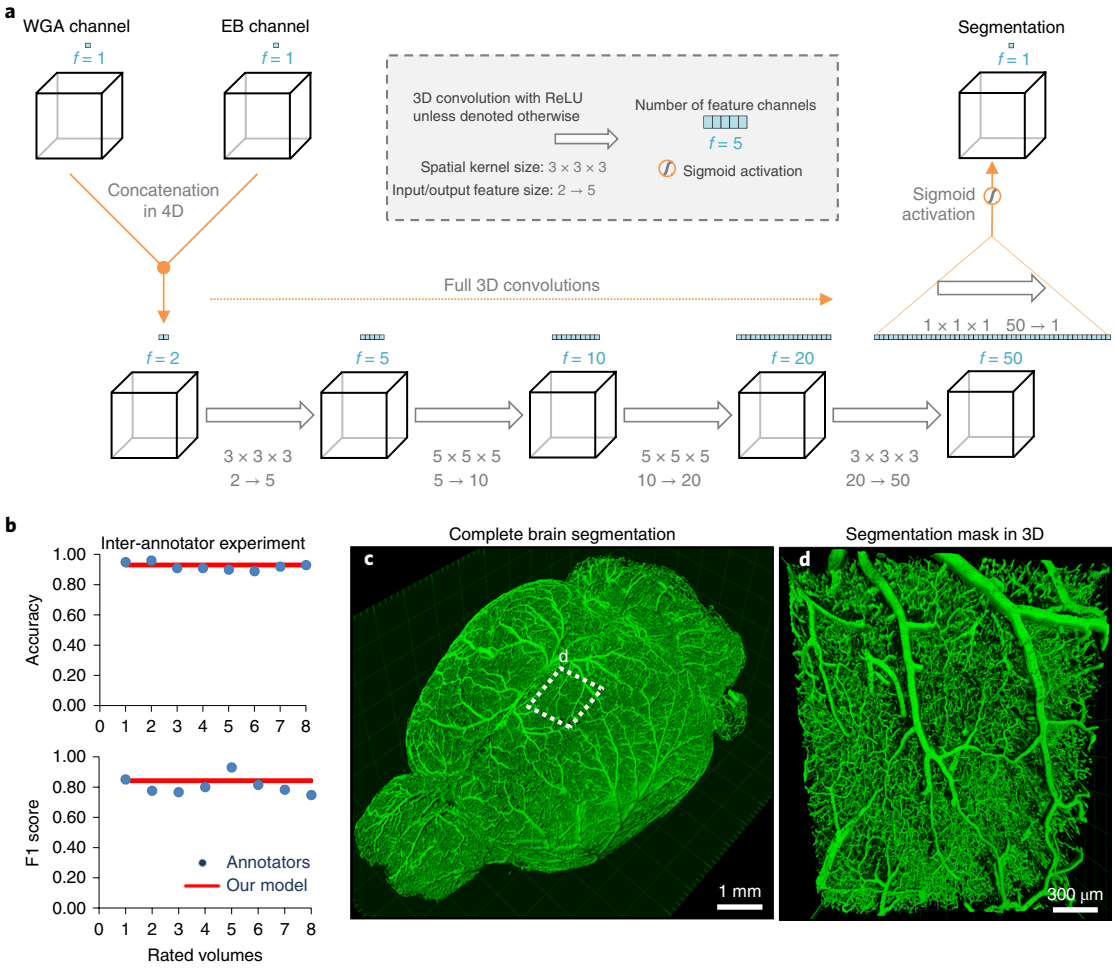


Fig. 3 | Deep learning architecture of VesSAP and performance on vessel segmentation. a, The 3D VesSAP network architecture consisting of five convolutional layers and sigmoid activation for the last layer, including the kernel size and feature size for the input/output. ReLU, rectified linear units. **b**, Accuracy and F1 score for the inter-annotator experiment (blue) as compared to VesSAP (red). **c**, 3D rendering of full brain segmentation from a CD1 mouse. **d**, 3D rendering of the small volume boxed in **c**. The experiment was performed on nine different mice with similar results.

Table 1 | Evaluation metrics of the different segmentation approaches for 75 volumes of 100 × 100 × 50 pixels

Segmentation model	cl-F1	Accuracy	F1 score	Jaccard	Parameters	Speed
VesSAP CNN	**0.93 ± 0.02***	**0.94 ± 0.01**	**0.84 ± 0.05**	**0.84 ± 0.04**	**0.0587 M***	**1.19 s***
VesSAP CNN, trained from scratch	**0.93 ± 0.02**	**0.94 ± 0.01**	**0.85 ± 0.04***	**0.85 ± 0.04**	**0.0587 M***	**1.19 s***
VesSAP CNN, synthetic training data	0.87 ± 0.02	0.90 ± 0.05	0.72 ± 0.07	0.70 ± 0.05	**0.0587 M***	**1.19 s***
3D U-Net	**0.93 ± 0.02**	**0.95 ± 0.01***	**0.85 ± 0.03***	**0.85 ± 0.03**	178.4537 M	61.22 s
V-Net	**0.94 ± 0.02***	**0.95 ± 0.02***	**0.86 ± 0.07***	**0.86 ± 0.07***	88.8556 M	26.87 s
Frangi vesselness	0.84 ± 0.03	0.85 ± 0.03	0.47 ± 0.19	-	-	117.00 s
Markov random field	0.86 ± 0.02	0.85 ± 0.03	0.48 ± 0.04	-	-	24.31 s

All values are given as the mean ± s.d. The best performing algorithms are in bold and highlighted with an asterisk; algorithms whose performance did not differ more than 2% from the best performing algorithms are in bold. The number of trainable parameters for deep learning architectures is given in millions (M).

V-Net and 3D U-Net, respectively. This is particularly important for our large datasets (hundreds of gigabytes). For example, the VesSAP CNN segmented a single brain in 4 h, whereas V-Net and 3D U-Net required 3.3 d and 8 d, respectively. The superior speed of the VesSAP CNN is due to the substantially fewer trainable parameters in its architecture (for example, our implementation of 3D U-Net had ~178 million parameters, whereas the VesSAP CNN had ~0.059 million parameters) (Table 1). Next, we compared the segmentation accuracy of our network to the accuracy of human annotations. A total of four human experts independently annotated two volumes. We found that the inter-annotator accuracy and F1 scores of the experts were comparable to those from the predicted segmentation of our network (human annotators: accuracy, 0.92 ± 0.02; F1 score, 0.81 ± 0.06; Fig. 3b). Notably, we extrapolate that human annotators

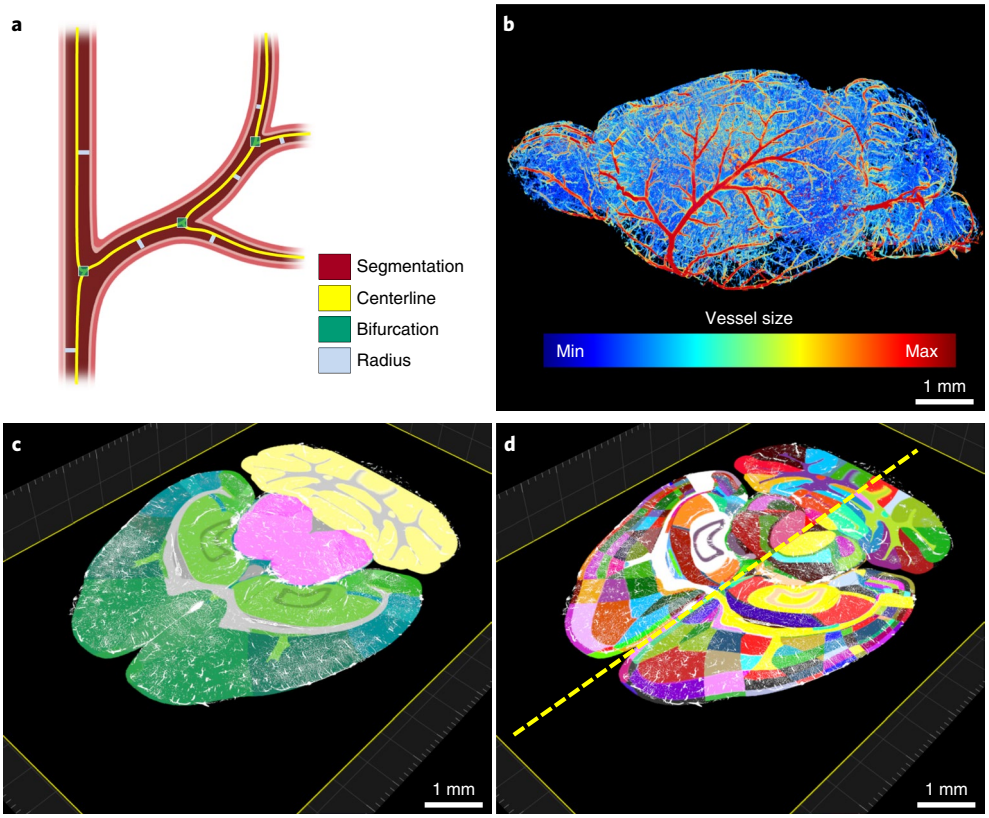


Fig. 4 | Pipeline showing the feature extraction and registration process. a, Representation of the features extracted from vessels. **b**, Radius illustration of the vasculature in a CD1 mouse brain. **c**,**d**, Vascular segmentation results overlaid on the hierarchically (**c**) and randomly (**d**) color-coded atlas to reveal all annotated regions available, including hemispheric difference (dashed line in **d**). The experiment was performed on nine different mice with similar results.

would need more than a year to process a whole brain instead of the 4 h required by our approach. Moreover, we observed differences in the human segmentations due to annotator bias. Thus, the VesSAP CNN can segment the complete brain vasculature consistently at human-level accuracy with a substantially higher speed than currently available methods, enabling high throughput for large-scale analysis.

We show an example of the vasculature from a brain segmented by VesSAP in 3D (Fig. 3c and Supplementary Videos 2 and 3). Zooming in on a smaller patch revealed that the connectivity of the vascular network was fully maintained (Fig. 3d and Supplementary Video 2). Comparing single slices of the imaging data with the predicted segmentation showed that vessels were accurately segmented regardless of absolute illumination or vessel diameter (Supplementary Fig. 7).

Feature extraction and atlas registration. Vessel lengths and radii and the number of bifurcation points are commonly used to describe the angioarchitecture[2]. Hence, we used our segmentation to quantify these features as distinct parameters to characterize the mouse brain vasculature (Fig. 4a and Supplementary Video 4). We evaluated the local vessel length (length normalized to the size of the brain region of interest), local bifurcation density (sum of the occurrences normalized to the size of the brain region of interest) and local vessel radius (average radius along the full length) of blood vessels in different brain regions.

We report the vascular features in three ways to enable comparison with various previous studies that differed in the measures used (Supplementary Fig. 8). More specifically, first, we provide the count of segmented voxels as compared to total voxels within a specific brain region (voxel space). Second, we provide the measurements by calculating the voxel size of our imaging system and accounting for the Euclidean length (microscopic space). Third, we corrected the microscopic measurements to account for tissue shrinkage caused by the clearing process (anatomical space)[35,36] (Supplementary Tables 2–10). We calculated this shrinkage rate by measuring the same mouse brain volume with MRI before clearing.

Here we use the anatomical space to report our specific biological findings, as it is closest to the physiological state. For the average blood vessel length of the whole brain, we found a value of 545.74 ± 94 mm per mm^3 (mean ± s.d.). Because our method quantifies brain regions separately, we could compare our results to the literature, which mostly reports either quantifications for specific brain regions or extrapolations to the whole brain from regional quantifications. For example, a vascular length of 922 ± 176 mm per mm^3 (mean ± s.d.) was previously reported for cortical regions (size of $508 \times 508 \times 1{,}500$ µm^3)[10]. We found a similar vessel length for the same region in the mouse cortex (C57BL/6J mice: 913 ± 110 mm per mm^3), substantiating the accuracy of our method. We performed additional comparisons to other reports (Supplementary Table 11). Moreover, we compared the measurements acquired with our algorithms to manually labeled ground truth data and found deviations of 8.21% for centerlines, 13.18% for the number of bifurcation points and 16.33% for the average radius. These deviations were substantially lower than the average deviation among human annotators (Methods).

We quantified and visualized vessel radius along the entire vascular network (Fig. 4b). After extracting vascular features for the whole brain with VesSAP, we registered the volume to the Allen brain atlas (Supplementary Videos 5 and 6). This allowed us to map the segmented vasculature and corresponding features topographically to distinct anatomical brain regions (Fig. 4c). Each anatomical

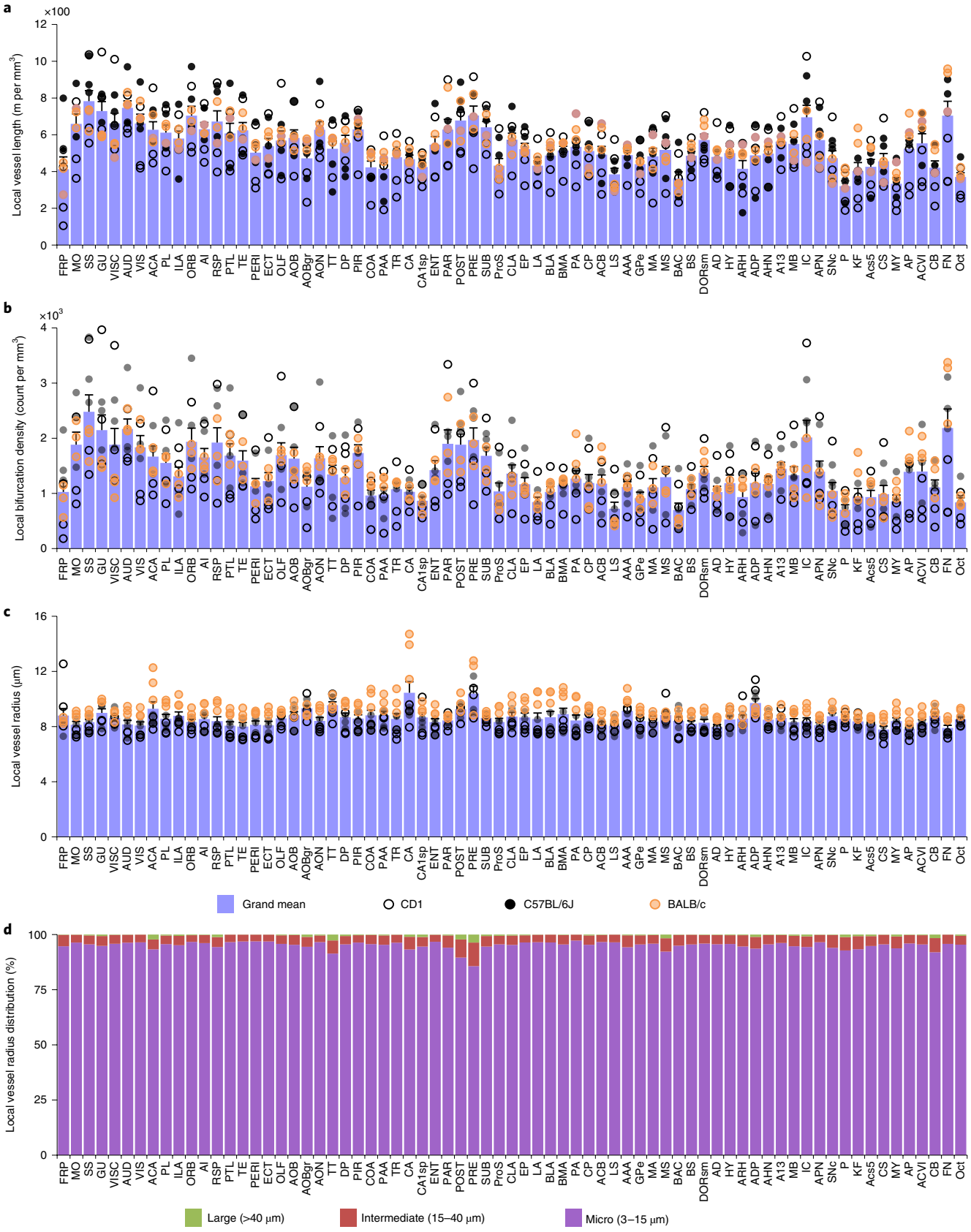


Fig. 5 | Anatomical properties of the neurovasculature in adult mouse brain mapped to the Allen brain atlas clusters. a–c, Representations of the local vessel length (**a**), density of bifurcations (**b**) and average radius (**c**) in each of the 71 main anatomical clusters of the Allen brain atlas. Open, black and orange circles denote measurements in the CD1, C57BL/6J and BALB/c strains, respectively; each circle represents a single mouse. Data are given as the mean ± s.e.m.; $n = 3$ mice per strain. **d**, Local distribution of large, intermediate and microvessels in the same anatomical clusters. Abbreviations are defined in Supplementary Table 1.

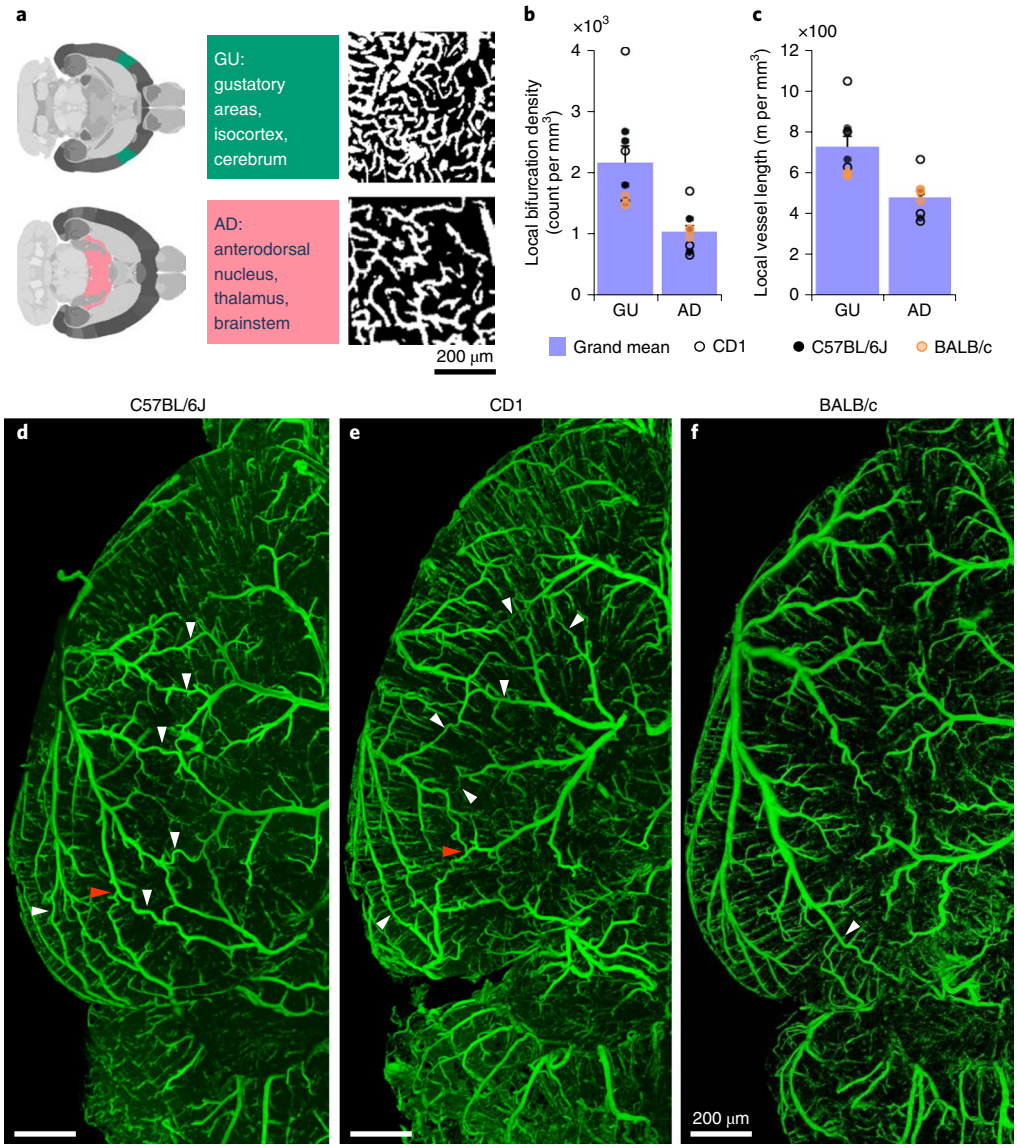


Fig. 6 | Exemplary quantitative analysis enabled by VesSAP. a, Respective locations of the anterodorsal nucleus (AD) and gustatory areas (GU) in the mouse brain (left) and maximum-intensity projections of representative volumes from segmentation of these areas ($600 \times 600 \times 33\,\mu m^3$) (right). **b**,**c**, Quantification of the bifurcation density (**b**) and local vessel length (**c**) for the anterodorsal nucleus and gustatory area clusters. CD1 mice are shown by open circles, BALB/C mice by orange circles and C57BL/6J mice by black circles. Values are the mean ± s.e.m.; $n = 3$ mice per strain. **d**–**f**, Images of the vasculature in representative C57BL/6J (**d**), CD1 (**e**) and BALB/c (**f**) mice, where white arrowheads indicate anastomoses between major arteries. Direct vascular connections between the medial cerebral artery, the anterior cerebral artery and the posterior cerebral artery are indicated by red arrowheads. The experiment was performed three times with similar results.

region could be further divided into subregions, yielding a total of 1,238 anatomical structures (619 per hemisphere) for the entire mouse brain (Fig. 4d). This allowed analysis of each denoted brain region and grouping of regions into clusters such as left versus right hemisphere, gray versus white matter, or any hierarchical cluster of the Allen brain atlas ontology. For our subsequent statistical feature analysis, we grouped the labeled structures according to the 71 main anatomical clusters of the current Allen brain atlas ontology. We thus provide the whole mouse brain vascular map with extracted vessel lengths, bifurcation points and radii down to the capillary level.

VesSAP provides a reference map of the whole brain vasculature in mice. By studying whole brain vasculature in the C57BL/6J, CD1 and BALB/c strains ($n = 3$ mice for each strain), we found that the local vessel length and local bifurcation density differed in the same brain over different regions, while they were highly correlated among different mice for the same regions (Fig. 5a,b). Furthermore, the local bifurcation density was highly correlated with the local vessel length in most brain regions (Supplementary Fig. 9), and the average vessel radius was evenly distributed in different regions of the same brain (Fig. 5c). In addition, the extracted features showed no statistical difference (by Cohen's *d*; Supplementary Table 12) for the same anatomical cluster across the strains (Supplementary Fig. 9). Finally, microvessels made up the overwhelming majority of the total vascular composition in all brain regions (Fig. 5d). We visually inspected exemplary brain regions to validate the output of VesSAP. Both VesSAP and visual inspection revealed that the gustatory areas had a higher vascular length per volume than the anterodorsal nucleus (Fig. 6a–c). Visual inspection also suggested that the number of capillaries was the primary reason for regional feature variations within the same brain.

Finally, VesSAP offered insights into the neurovascular structure of the different mouse strains in our study. There were direct intracranial vascular anastomoses in the C57BL/6J, CD1 and BALB/c strains (white arrowheads in Fig. 6d–f). The anterior cerebral artery, middle cerebral artery and posterior cerebral artery were connected at the dorsal visual cortex in CD1 mice (red arrowheads in Fig. 6d,e) unlike in the BALB/c strain[33] (Fig. 6f).

Discussion

VesSAP can generate reference maps of the adult mouse brain vasculature, which can potentially be used to model synthetic cerebrovascular networks[37]. In addition to the metrics we obtain to describe the vasculature, advanced metrics, for example, Strahler values, network connectivity and bifurcation angles, can be extracted by using the data generated by VesSAP. Furthermore, the centerlines and bifurcation points can be interpreted as the edges and nodes for building a full vascular network graph, offering a means for studying local and global properties of the cerebrovascular network in the future.

The VesSAP workflow relies on staining of blood vessels by two different dyes. WGA binds to the glycocalyx of the endothelial lining of blood vessels[38] but may miss some segments of large vessels[18]. EB is a dye with a high affinity for serum albumin[35,36,39]; thus, it remains in the large vessels after a short perfusion protocol. In addition, EB labeling is not affected by subsequent DISCO clearing.

Vessels have long and thin tubular shapes. In our images, the radii of capillaries (about 3 µm) are in the range of our voxel size. Therefore, segmentation that yields the correct diameter down to single-pixel resolution poses a challenge, as we observed a 16% deviation for the radius. This subpixel deviation did not pose a problem for segmenting the whole vasculature network and extracting features because the vascular network can be defined by its centerlines and bifurcations.

The described segmentation concept is based on a transfer learning approach, where we pretrained the CNN and refined it on a small labeled dataset of 11% of the synthetic dataset and only 0.02% of one cleared brain. We consider this to be a major advantage in comparison to training from scratch. Thus, our CNN might generalize well to different types of imaging data (such as micro-CT angiography) or other curvilinear structures (for example, neurons), as only a small labeled dataset is needed to adjust our pretrained network.

On the basis of our vascular reference map, unknown vascular properties can be discovered and biological models can be confirmed. VesSAP showed a high number of collaterals in albino CD1 mice. Such collaterals between large vessels can substantially alter the outcome of ischemic stroke lesions: blood-deprived brain regions arising from occlusion of a large vessel can be compensated by blood supply from the collateral extensions of other large vessels[33,40]. Therefore, our VesSAP method can lead to the discovery of previously unknown anatomical details that could be functionally relevant.

In conclusion, VesSAP is a scalable, modular and automated machine learning-based method to analyze complex imaging data from cleared mouse brains. We foresee that our method will accelerate the applications of tissue clearing, in particular for studies assessing brain vasculature.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-020-0792-1.

Received: 19 April 2019; Accepted: 14 February 2020;
Published online: 11 March 2020

References

1. Bennett, R. E. et al. Tau induces blood vessel abnormalities and angiogenesis-related gene expression in P301L transgenic mice and human Alzheimer's disease. *Proc. Natl Acad. Sci. USA* **115**, E1289–E1298 (2018).
2. Obenaus, A. et al. Traumatic brain injury results in acute rarefication of the vascular network. *Sci. Rep.* **7**, 239 (2017).
3. Völgyi, K. et al. Chronic cerebral hypoperfusion induced synaptic proteome changes in the rat cerebral cortex. *Mol. Neurobiol.* **55**, 4253–4266 (2018).
4. Klohs, J. et al. Contrast-enhanced magnetic resonance microangiography reveals remodeling of the cerebral microvasculature in transgenic ArcAbeta mice. *J. Neurosci.* **32**, 1705–1713 (2012).
5. Edwards-Richards, A. et al. Capillary rarefaction: an early marker of microvascular disease in young hemodialysis patients. *Clin. Kidney J.* **7**, 569–574 (2014).
6. Calabrese, E., Badea, A., Cofer, G., Qi, Y. & Johnson, G. A. A diffusion MRI tractography connectome of the mouse brain and comparison with neuronal tracer data. *Cereb. Cortex* **25**, 4628–4637 (2015).
7. Dyer, E. L. et al. Quantifying mesoscale neuroanatomy using X-ray microtomography. *eNeuro* **4**, ENEURO.0195-17.2017 (2017).
8. Li, T., Liu, C. J. & Akkin, T. Contrast-enhanced serial optical coherence scanner with deep learning network reveals vasculature and white matter organization of mouse brain. *Neurophotonics* **6**, 035004 (2019).
9. Tsai, P. S. et al. Correlations of neuronal and microvascular densities in murine cortex revealed by direct counting and colocalization of nuclei and vessels. *J. Neurosci.* **29**, 14553–14570 (2009).
10. Lugo-Hernandez, E. et al. 3D visualization and quantification of microvessels in the whole ischemic mouse brain using solvent-based clearing and light sheet microscopy. *J. Cereb. Blood Flow Metab.* **37**, 3355–3367 (2017).
11. Frangi, A. F., Niessen, W. J., Vincken, K. L. & Viergever, M. A. in *International Conference on Medical Image Computing and Computer-Assisted Intervention* 130–137 (Springer, 1998).
12. Sato, Y. et al. Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Med. Image Anal.* **2**, 143–168 (1998).
13. Di Giovanna, A. P. et al. Whole-brain vasculature reconstruction at the single capillary level. *Sci. Rep.* **8**, 12573 (2018).
14. Xiong, B. et al. Precise cerebral vascular atlas in stereotaxic coordinates of whole mouse brain. *Front. Neuroanat.* **11**, 128 (2017).
15. Clark, T. A. et al. Artery targeted photothrombosis widens the vascular penumbra, instigates peri-infarct neovascularization and models forelimb impairments. *Sci. Rep.* **9**, 2323 (2019).
16. Yao, J., Maslov, K., Hu, S. & Wang, L. V. Evans blue dye—enhanced capillary-resolution photoacoustic microscopy in vivo. *J. Biomed. Opt.* **14**, 054049 (2009).
17. Ertürk, A. et al. Three-dimensional imaging of solvent-cleared organs using 3DISCO. *Nat. Protoc.* **7**, 1983–1995 (2012).
18. Reitsma, S. et al. Endothelial glycocalyx structure in the intact carotid artery: a two-photon laser scanning microscopy study. *J. Vasc. Res.* **48**, 297–306 (2011).
19. Schimmenti, L. A., Yan, H. C., Madri, J. A. & Albelda, S. M. Platelet endothelial cell adhesion molecule, PECAM-1, modulates cell migration. *J. Cell. Physiol.* **153**, 417–428 (1992).
20. Tetteh, G. et al. DeepVesselNet: vessel segmentation, centerline prediction, and bifurcation detection in 3-D angiographic volumes. Preprint at https://arxiv.org/abs/1803.09340 (2018).
21. Weigert, M. et al. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nat. Methods* **15**, 1090–1097 (2018).
22. Wang, H. et al. Deep learning enables cross-modality super-resolution in fluorescence microscopy. *Nat. Methods* **16**, 103–110 (2019).
23. Falk, T. et al. U-Net: deep learning for cell counting, detection, and morphometry. *Nat. Methods* **16**, 67–70 (2019).
24. Caicedo, J. C. et al. Data-analysis strategies for image-based cell profiling. *Nat. Methods* **14**, 849–863 (2017).
25. Dorkenwald, S. et al. Automated synaptic connectivity inference for volume electron microscopy. *Nat. Methods* **14**, 435–442 (2017).
26. Liu, S., Zhang, D., Song, Y., Peng, H. & Cai, W. in *Machine Learning in Medical Imaging* (eds., Wang, Q. et al.) 185–193 (Springer International Publishing, 2017).
27. Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 3431–3440 (IEEE, 2015).
28. Livne, M. et al. A U-Net deep learning framework for high performance vessel segmentation in patients with cerebrovascular disease. *Front. Neurosci.* **13**, 97 (2019).
29. Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T. & Ronneberger, O. 3D U-Net: learning dense volumetric segmentation from sparse annotation. in *Medical Image Computing and Computer-Assisted Intervention* 424–432 (2016).
30. Milletari, F., Navab, N. & Ahmadi, S.A. V-Net: fully convolutional neural networks for volumetric medical image segmentation. in *2016 Fourth International Conference on 3D Vision* 565–571 (IEEE, 2016).

31. Litjens, G. et al. A survey on deep learning in medical image analysis. *Med. Image Anal.* **42**, 60–88 (2017).
32. Schneider, M., Reichold, J., Weber, B., Szekely, G. & Hirsch, S. Tissue metabolism driven arterial tree generation. *Med. Image Anal.* **16**, 1397–1414 (2012).
33. Chalothorn, D., Clayton, J. A., Zhang, H., Pomp, D. & Faber, J. E. Collateral density, remodeling, and VEGF-A expression differ widely between mouse strains. *Physiol. Genomics* **30**, 179–191 (2007).
34. Li, S. Z. In *Computer Vision—ECCV* (ed., Eklundh, J.-O.) 361–370 (Springer, 1994).
35. Pan, C. et al. Shrinkage-mediated imaging of entire organs and organisms using uDISCO. *Nat. Methods* **13**, 859–867 (2016).
36. Cai, R. et al. Panoptic imaging of transparent mice reveals whole-body neuronal projections and skull–meninges connections. *Nat. Neurosci.* **22**, 317–327 (2019).
37. Menti, E., Bonaldi, L., Ballerini, L., Ruggeri, A. & Trucco, E. In *International Workshop on Simulation and Synthesis in Medical Imaging* 167–176 (Springer, Year).
38. Kataoka, H. et al. Fluorescent imaging of endothelial glycocalyx layer with wheat germ agglutinin using intravital microscopy. *Microsc. Res. Tech.* **79**, 31–37 (2016).
39. Steinwall, O. & Klatzo, I. Selective vulnerability of the blood–brain barrier in chemically induced lesions. *J. Neuropathol. Exp. Neurol.* **25**, 542–559 (1966).
40. Zhang, H., Prabhakar, P., Sealock, R. & Faber, J. E. Wide genetic variation in the native pial collateral circulation is a major determinant of variation in severity of stroke. *J. Cereb. Blood Flow Metab.* **30**, 923–934 (2010).

Methods

Tissue preparation. Animal experiments were conducted according to institutional guidelines (Klinikum der Universität München/Ludwig Maximilian University of Munich), after approval of the ethical review board of the government of Upper Bavaria (Regierung von Oberbayern, Munich, Germany), and in accordance with European directive 2010/63/EU for animal research. Animals were housed under a 12-h light/12-h dark cycle. For this study, we injected 150 µl (2% (vol/vol) in saline) EB dye (Sigma-Aldrich, E2129) intraperitoneally into 3-month-old male mice from the C57BL/6J, CD1 and BALB/c strains (Charles River, strain codes 027, 482 and 028, respectively; $n=3$ mice per strain). Twelve hours after injection of EB dye, we anesthetized the animals with a combination of midazolam, medetomidine and fentanyl (administered intraperitoneally; 1 ml per 100 g body weight containing 5 mg, 0.5 mg and 0.05 mg per kg body weight, respectively) and opened their chest for transcardial perfusion. Medium with WGA (0.25 mg WGA conjugated to Alexa Fluor 594 dye (Thermo Fisher Scientific, W11262) in 150 µl PBS, pH 7.2) was supplied by peristaltic pump set to deliver the medium at a rate of 8 ml min^{-1}, along with 15 ml of 1× PBS and 15 ml of 4% paraformaldehyde. This short perfusion protocol was established on the basis of preliminary experiments, where both WGA and EB staining were partially washed out (data not shown), with the goal of delivering fixative to brain tissue via the vessels to achieve a homogenous preservation effect[41].

After perfusion, brains were extracted from the neurocranium while severing some of the segments of the circle of Willis, which is an inevitable component of most retrieval processes aside from corrosion cast techniques. Next, the samples were incubated in 3DISCO clearing solutions as described[17]. Briefly, we immersed them in a gradient of tetrahydrofuran (Sigma-Aldrich, 186562): 50%, 70%, 80% and 90% (in distilled water) followed by 100%, at 25 °C for 12 h at each concentration. At this point, we modified the protocol by incubating the samples in *tert*-butanol for 12 h at 35 °C followed by immersion in dichloromethane (Sigma-Aldrich, 270997) for 12 h at room temperature and a final incubation with refractive index-matched BABB solution (benzyl alcohol + benzyl benzoate, 1:2; Sigma-Aldrich, 24122 and W213802), for at least 24 h at room temperature until transparency was achieved. Each incubation step was carried out on a laboratory shaker.

Imaging of cleared samples with light-sheet microscopy. We used a ×4 objective lens (Olympus XLFLUOR 340) equipped with an immersion-corrected dipping cap mounted on a LaVision UltraII microscope coupled to a white-light laser module (NKT SuperK Extreme EXW-12) for imaging. Images were taken with 16-bit depth and at a nominal resolution of 1.625 µm per voxel on the *x* and *y* axes. For ×12 imaging, we used a LaVision objective (×12/0.53 NA MI PLAN with an immersion-corrected dipping cap). Brain structures were visualized by Alexa Fluor 594 (using a 580/25-nm excitation filter and a 625/30-nm emission filter) and EB fluorescent dye (using a 640/40-nm excitation filter and a 690/50-nm emission filter) in sequential order. We maximized the SNR for each dye independently to avoid saturation of differently sized vessels when only a single channel was used. We achieved this by independently optimizing the excitation power so that the strongest signal in major vessels did not exceed the dynamic range of the camera. In the *z* dimension, we took sectional images in 3-µm steps while using left- and right-sided illumination. Our measured resolution was 2.83 µm × 2.83 µm × 4.99 µm for *x*, *y* and *z*, respectively (Supplementary Fig. 2). To reduce defocus, which derives from the Gaussian shape of the beam, we used 12-step sequential shifting of the focal position of the light sheet per plane and side. The thinnest point of the light sheet was 5 µm.

Imaging of cleared samples with confocal microscopy. Additionally, the cleared specimens were imaged with an inverted laser-scanning confocal microscope (Zeiss, LSM 880) for further analysis. Before imaging, samples were mounted by placing them onto the glass surface of a 35-mm glass-bottom Petri dish (MatTek, P35G-0-14-C) and immersed in BABB. A ×40 oil-immersion objective lens was used (Zeiss, ECPlan-NeoFluar ×40/1.30 NA Oil DIC M27, WD = 0.21 mm). Images were acquired with the settings for Alexa Fluor 594 (using excitation at 561 nm and an emission range of 585–733 nm) and EB fluorescent dye (using excitation at 633 nm and an emission range of 638–755 nm) in sequential order.

Magnetic resonance imaging. We used a nanoScan PET/MR device (3 Tesla, Mediso Medical Imaging Systems) equipped with a head coil for murine heads to acquire anatomical scans in the T1 modality.

Reconstruction of the datasets from tiling volumes. We stitched the acquired volumes by using TeraStitcher's automatic global optimization function (v1.10.3). We produced volumetric intensity images of the whole brain while considering each channel separately. To improve alignment to the Allen brain atlas, we downscaled our dataset in the *xy* plane to achieve pseudouniform voxel spacing matching the *z* plane.

Deep learning network architecture. We relied on a deep 3D CNN for segmentation of our blood vessel dataset. The network's general architecture consists of five convolutional layers, four with ReLU (rectified linear units) followed by one convolutional layer with sigmoid activation (Fig. 3a). The input layer is designed to take *n* images as input. In the implemented case, the input to the first layer of the network comprised $n=2$ images of the same brain, which had been stained differently (Fig. 3a). To specifically account for the general class imbalance (much more tissue background than vessel signal) in our dataset and the potential for high false-positive rates associated with this, we chose the generalized soft-Dice as the loss function to our network. At all levels, we used full 3D convolutional kernels (Fig. 3a).

The network's training is driven by an Adam optimizer with a learning rate of 1×10^{-5} and an exponential decay rate of 0.9 for the first moment and 0.99 for the second moment[42]. A prediction or segmentation with a trained model takes volumetric images of arbitrary size as input and outputs a probabilistic segmentation map of identical size. To deal with volumes of arbitrary size and extension, we processed them in smaller subvolumes of 100 × 100 × 50 pixels in size. The algorithms were implemented by using the Tensorflow framework and KERAS[43]. They were trained and tested on two NVIDIA Quadro P5000 GPUs and on machines with 64 GB and 512 GB of RAM.

Transfer learning. Typically, supervised learning tasks in biomedical imaging are aggravated by the scarce availability of labeled training data. Our transfer learning approach aims to circumvent this problem by pretraining our models on synthetically generated data and refining them on a small set of real images[44]. Specifically, our approach pretrains the VesSAP CNN on 3D volumes of vascular image data, synthetically generated together with the corresponding training labels by using the approach of Schneider and colleagues[45]. The pretraining is carried out on a dataset of 20 volumes of 325 × 304 × 600 pixels in size for 38 epochs. During pretraining, we applied a learning rate of 1×10^{-4}. Afterward, the pretrained model was fine-tuned by retraining on a real microscopic dataset consisting of 11 volumes of 500 × 500 × 50 pixels in size. The image volumes were manually annotated by commissioned experts, including the expert who previously prepared the samples and operated the microscope. The labels were verified and further refined in consensus by two additional human raters. The data we used in this fine-tuning step amounted to 11% of the volume of the synthetic datasets and only 0.02% of the voxel volume of a single whole brain. For the fine-tuning step, we used a learning rate of 1×10^{-5}. The final model was obtained after training on the real dataset for six epochs. This training was substantially shorter than training from scratch, where we trained the same VesSAP CNN architecture for 72 epochs until we reached the best F1 score on the validation set. The labeled dataset consisted of 17 volumes of 500 × 500 × 50 pixels from five mouse brains. Three of these brains were from the CD1 strain, and two were from the C57BL/6J strain. The volumes were chosen from regions throughout the whole brain, to represent the variability in the vascular dataset in terms of both vessel shape and illumination. To ensure independence, volumes for the training set and test/validation set were chosen from independent brains. All datasets included brains from the two strains. Our training dataset consisted of 11 volumes, the validation dataset of 3 volumes and the test dataset of 3 volumes. We cross-tested on our test and validation datasets by rotating these. The volumes were processed during training and inference in 25 small subvolumes of 100 × 100 × 50 pixels.

We observed an average F1 score of 0.84 ± 0.02 (mean ± s.d.), an average accuracy of 0.94 ± 0.01 (mean ± s.d.) and an average Jaccard coefficient of 0.84 ± 0.04 (mean ± s.d.) on our test datasets (Fig. 3b). We tested the statistical significance of differences among the top three learning methods (the VesSAP CNN, V-Net and 3D U-Net) by using paired *t* tests. We found that the differences in F1 score were not statistically significant (all $P>0.3$, rejecting the hypothesis of different distributions).

Because the F1 score, accuracy and Jaccard coefficient are all voxel-wise volumetric scores and can fall short in evaluating the connectedness of components, we developed the cl-F1 score. cl-F1 is calculated from the intersection of centerlines and vessel volumes and not from volumes only, as the traditional F1 score is[46]. To determine this score, we first calculated the intersection of the centerline of our prediction with the labeled volume and then calculated the intersection of the labeled volume's centerline with the predicted volume. Next, we treated the first intersection as recall, as it is susceptible to false negatives, and the second intersection as precision, as it is susceptible to false positives, and input this into the traditional F1 score formulation:

$$\mathrm{F1} = 2\times\frac{\mathrm{precision}\times\mathrm{recall}}{\mathrm{precision}+\mathrm{recall}} \quad (1)$$

We report an average cl-F1 score of 0.93 ± 0.02 (mean ± s.d.) on the test set.

All scores are given as mean and s.d. Our model reached the best model selection point on the validation dataset after six epochs of training.

Comparison to 3D U-Net and V-Net. To compare our proposed architecture to different segmentation architectures, we implemented V-Net and 3D U-Net, both of which use more complex CNNs with substantially more trainable parameters, which further include down- and upsampling. While our experiments showed that 3D U-Net and V-Net reached marginally higher performance scores, the differences were not statistically significant (two-sided *t* test, $P>0.3$). The amount

of parameters for these tools makes them 51 and 23 times slower than VesSAP during the inference stage. For segmentation of one of our large whole-brain datasets, this translated to 4 h for VesSAP versus 8 d for 3D U-Net and 3.8 d for V-Net. This difference was also prevalent in the number of trainable parameters. The VesSAP CNN had 0.058 million parameters, whereas 3D U-Net consisted of more than 178 million and V-Net of more than 88 million parameters. Furthermore, the light VesSAP CNN already reached human-level performance. We therefore consider the problem of vessel segmentation as solved by the VesSAP CNN for our data. It should be mentioned that the segmentation network is a modular building block of the overall VesSAP pipeline and can be chosen by each user according to his or her own preferences and, importantly, according to the computational power available.

Preprocessing of segmentation. Preprocessing factors into the overall success of the training and segmentation. The intensity distribution among brains and among brain regions differs substantially. To account for intensity distributions, two preprocessing strategies were applied successively.

1. High-cut filter. In this step, the intensities x above a certain threshold c are set to c; c is defined by a global percentile. Next, they were normalized by $f(x)$.

$$g(x) = \begin{cases} c, & x > c \\ x, & x \leq c \end{cases} \quad (2)$$

2. Normalization of intensities. The original intensities were normalized to a range of 0 to 1, where x was the pixel intensity and X was all intensities for the volume.

$$f(x) = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (3)$$

Inter-annotator experiment for segmentation. To compare VesSAP's segmentation to human-level annotations, we implemented an inter-annotator experiment. For this experiment, we determined a gold-standard label (ground truth) for two volumes of 500×500×50 pixels from a commissioned group of three experts, including the expert who imaged our data and was therefore most familiar with the images. Next, we gave the two volumes to four other experts to label the complete vasculature. The experts spent multiple hours labeling each patch in the ImageJ and ITK-snap environment and were allowed to use their favored approaches to generate what they considered to be the most accurate labeling. Finally, we calculated the accuracy and F1 scores for the different annotators, as compared to the gold standard, and compared them to the scores for our model (Table 1).

Feature extraction. To quantify the anatomy of the mouse brain vasculature, we extracted descriptive features on the basis of our segmentation. First, we calculated the features in voxel space. Later, we registered them to the Allen brain atlas.

As features we extracted the centerlines, the bifurcation points and the radii of the segmented blood vessels. We consider these features to be independent from the elongation of the light-sheet scans and the connectedness of the vessels due to staining, imaging and/or segmentation artifacts.

Our centerline extraction was based on a 3D thinning algorithm[47]. Before extracting the centerlines, we applied two cycles of binary erosion and dilation to remove false-negative pixels within the volume of segmented vessels, as these would induce false centerlines. On the basis of the centerlines, we extracted bifurcation points. A bifurcation was the branching point on a centerline where a larger vessel split into two or more smaller vessels (Fig. 4a). In a network analysis context, bifurcations are meaningful as they represent the nodes of a vascular network[48]. Furthermore, bifurcation points have relevance in a biological context. In neurodegenerative diseases, capillaries are known to degenerate[49], thereby substantially reducing the number of bifurcation points in an affected brain region as compared to healthy brain. Next, we implemented an algorithm to detect bifurcation points. We achieved this by calculating the surrounding pixels for every point on each centerline and determined whether a point was a centerline. The radius of a blood vessel is a key feature to describe vascular networks. The radius yields information about the flow and hierarchy of the vessel network. The proposed algorithm calculates the Euclidean distance transform for every segmented pixel v to the closest background pixel b_{closest}. Next, the distance transform matrix is multiplied by the 3D centerline mask, equaling the minimum radius of the vessel around the centerline.

$$d(v, b_{\text{closest}}) = \sqrt{\sum_{1}^{3} (v_i - b_{\text{closest},i})^2} \quad (4)$$

Feature quantification. Here we describe in detail how we calculated the features between the three different spaces.

Voxel space to microscopic space. To quantify the length of our vessels in SI units instead of voxels, we calculated their Euclidean length, which depends on the direction of the connection of skeleton pixels (Supplementary Fig. 9). To calculate the Euclidean length of our centerlines, we carried out a connected component analysis, which transformed each pixel of the skeleton into an element of an undirected weighted graph, where zero weight means no connection and non-zero weights denote the Euclidean distance between two voxels (considering 26 connectivity). Thus, we obtained a large and sparse adjacency matrix. An element-wise summation of such a matrix provides the total Euclidean length of the vascular network along the extracted skeleton.

As measuring connected components is computationally very expensive, we calculated the Euclidean length of the centerlines for 12 representative volumes of 500×500×50 pixels and divided by the number of skeleton pixels. We calculated an average Euclidean length ε_{Cl} of 1.3234 ± 0.0063 voxels (mean ± s.d.) per centerline element. This corresponds to a length of 3.9701 ± 0.0188 µm (mean ± s.d.) in cleared tissue. Because the s.d. of this measurement was low, at less than 0.5% of the length, we applied this correction factor to the whole brain centerline measurements. This correction does not apply to the bifurcation points and our radius statistics, as bifurcations are independent of length and also radius extraction returns a Euclidean distance by default. Depending on the direction of the connection of skeleton pixels, the Euclidean length of a skeleton pixel is different (Supplementary Fig. 9).

Microscopic space to anatomical space. To account for tissue shrinkage (Supplementary Fig. 9), which is inherent to DISCO clearing, we carried out an experiment to measure the degree of shrinkage. Before clearing, we imaged the brains of three live BALB/c mice by MRI and calculated each brain's average volume, through precise manual segmentation by an expert. Next, we cleared three BALB/c brains, processed them with VesSAP and measured the total brain volume with atlas alignment. We report an average volume of 423.84 ± 2.04 mm^3 for the live mice and 255.62 ± 6.57 mm^3 for the cleared tissue. This corresponds to a total volume shrinkage of 39.69%. We applied this as a correction factor for the volumetric information (for example, for brain regions).

Similarly to previous studies, shrinkage was uniform in all three dimensions. This is important when considering shrinkage in one dimension, as needed to account for the shrinkage in centerlines and radii. The one-dimensional correction factor K_{L} then corresponds to the cube root of the volumetric correction factor K_{V}.

Accounting for these factors, we calculated the vessel length per volume (Z) in cleared (Z_{cleared}) and real (Z_{real}) tissue in equation (5), where $N_{\text{V,vox}}$ is the number of total voxels in the reference volume and $N_{\text{Cl,vox}}$ is the number of centerline voxels in the image volume:

$$Z_{\text{cleared}} = \frac{N_{\text{Cl,vox}}}{N_{\text{V,vox}}} \times \varepsilon_{\text{Cl}} \quad Z_{\text{real}} = \frac{N_{\text{Cl,vox}}}{N_{\text{V,vox}}} \times \varepsilon_{\text{Cl}} \times \frac{\kappa_{\text{L}}}{\kappa_{\text{V}}} \quad (5)$$

Similarly, we calculated the bifurcation density (B) in cleared and real tissue in equation (6), where $N_{\text{Bif,vox}}$ is the number of bifurcations in the reference volume:

$$B_{\text{cleared}} = \frac{N_{\text{Bif,vox}}}{N_{\text{V,vox}}} \quad B_{\text{real}} = \frac{N_{\text{Bif,vox}}}{N_{\text{V,vox}}} \times \frac{1}{\kappa_{\text{V}}} \quad (6)$$

Please note that the voxel spacing of 3 µm has to be taken into consideration when reporting features in SI units.

Inter-annotator experiment for features. To estimate the error in VesSAP's feature quantification, we extracted the features on a labeled test set of five volumes of 500×500×50 pixels. When comparing to the gold-standard label, we calculated errors (disagreements) of 8.21% for the centerlines, 13.18% for the number of bifurcation points and 16.33% for the average radius. To compare VesSAP's extracted features to human-level annotation, we implemented an inter-annotator experiment. For this experiment, we had four annotators label the vessels and radii in two volumes of 500×500×50 pixels by using ImageJ and ITK-snap. Finally, we calculated the agreement of the extracted features between all annotators and compared to the gold-standard labeling.

We calculated this for each of the volumes and found an average error (disagreement) of 34.62% for the radius, 25.20% for the bifurcation count and 12.55% for the centerline length.

The agreement between the VesSAP output and the gold standard was higher than the average agreement between the annotators and the gold standard. This difference underlines the quality and reproducibility of VesSAP's feature extraction.

Registration to the reference atlas. We used the average template, the annotation file and the latest ontology file (Ontology ID: 1) of the current Allen mouse brain atlas: CCFv3 201710. Then, we scaled the template and the annotation file up from 10 to 3 μm^3 to match our reconstructed brain scans and multiplied the left side of the (still symmetrical) annotation file by −1 so that the labels could be later assigned to the corresponding hemispheres. Next, the average template and 3D vascular datasets were downsampled to 10% of their original size in each dimension to achieve reasonably fast alignment with the elastix toolbox[50] (v4.9.0). For the sake of the integrity of the extracted features, we aligned the template to each of the brain scans individually by using a two-step rigid and deformable registration (B-spline; optimizer, AdaptiveStochasticGradientDescent; metric, AdvancedMattesMutualInformation; grid spacing in physical units, 90; in the VesSAP repository, we host the log and parameter files for each brain scan) and applied the transformation parameters to the full-resolution annotation volume

(3-µm resolution). Subsequently, we created masks for the anatomical clusters on the basis of the current Allen brain atlas ontology.

Statistical analysis of features. Data collection and analysis were not performed with blinding to the strains. Data distribution was assumed to be normal, although this was not formally tested. All data values of the descriptive statistics are given as mean ± s.e.m. unless stated otherwise. Data were analyzed with standardized effect size indices (Cohen's *d*)[51] to investigate differences in vessel length, number of bifurcation points and radii between brain areas across the three mouse strains ($n = 3$ mice per strain). Descriptive statistics were evaluated across brain regions in the pooled ($n = 9$) dataset.

Data visualization. All volumetric datasets were rendered with Imaris, Vision4D and ITK-snap.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

VesSAP data are publicly hosted at http://DISCOtechnologies.org/VesSAP and include original scans and registered atlas data.

Code availability

VesSAP codes are publicly hosted at http://DISCOtechnologies.org/VesSAP and include the imaging protocol, trained algorithms, training data and a reference set of features describing the vascular network in all brain regions. Additionally, the source code is hosted on GitHub (https://github.com/vessap/vessap) and on the executable platform Code Ocean (https://doi.org/10.24433/CO.1402016.v1)[52]. Implementation of external libraries is available on request.

References

41. Breckwoldt, M. O. et al. Correlated magnetic resonance imaging and ultramicroscopy (MR-UM) is a tool kit to assess the dynamics of glioma angiogenesis. *eLife* **5**, e11712 (2016).
42. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2017).
43. Abadi, M. et al. in *12th USENIX Symposium on Operating Systems Design and Implementation* 265–283 (2016).
44. Hoo-Chang, S. et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **35**, 1285–1298 (2016).
45. Schneider, M., Hirsch, S., Weber, B., Székely, G. & Menze, B. H. Joint 3-D vessel segmentation and centerline extraction using oblique Hough forests with steerable filters. *Med. Image Anal.* **19**, 220–249 (2015).
46. Paetzold, J. C. et al. clDice—a novel connectivity-preserving loss function for vessel segmentation. in *Medical Imaging Meets NeurIPS 2019 Workshop* (2019).
47. Lee, T. C., Kashyap, R. L. & Chu, C. N. Building skeleton models via 3-D medial surface axis thinning algorithms. *Graph. Models Image Process.* **56**, 462–478 (1994).
48. Rempfler, M. et al. Reconstructing cerebrovascular networks under local physiological constraints by integer programming. *Med. Image Anal.* **25**, 86–94 (2015).
49. Marchesi, V. T. Alzheimer's dementia begins as a disease of small blood vessels, damaged by oxidative-induced inflammation and dysregulated amyloid metabolism: implications for early detection and therapy. *FASEB J.* **25**, 5–13 (2011).
50. Klein, S., Staring, M., Murphy, K., Viergever, M. A. & Pluim, J. P. Elastix: a toolbox for intensity-based medical image registration. *IEEE Trans. Med. Imaging* **29**, 196–205 (2009).
51. Cohen, J. The effect size index: *d*. *Stat. Power Anal. Behav. Sci.* **2**, 284–288 (1988).
52. Paetzold, J. C. & Tetteh, G. VesSAP: machine learning analysis of whole mouse brain vasculature. *Code Ocean* https://doi.org/10.24433/CO.1402016.v1 (2020).

Acknowledgements

This work was supported by the Vascular Dementia Research Foundation, Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy within the framework of the Munich Cluster for Systems Neurology (EXC 2145 SyNergy, ID 390857198), ERA-Net Neuron (01EW1501A to A.E.), Fritz Thyssen Stiftung (to A.E.; reference no. 10.17.1.019MN), a DFG research grant (to A.E.; reference no. ER 810/2-1), the Helmholtz ICEMED Alliance (to A.E.), the NIH (to A.E.; reference no. AG057575) and the German Federal Ministry of Education and Research via the Software Campus initiative (to O.S.). S.S. is supported by the Translational Brain Imaging Training Network (TRABIT) under the European Union's Horizon 2020 research and innovation program (grant agreement ID 765148). Furthermore, NVIDIA supported this work via the GPU Grant Program. V.E. was funded by Human Brain Project (HBP SGA 2, 785907). M.I.T. is a member of the Graduate School of Systemic Neurosciences (GSN), Ludwig Maximilian University of Munich. We thank R. Cai, C. Pan, F. Voigt, I. Ezhov, A. Sekuboyina, M. Goergens, F. Hellal, R. Malik, U. Schillinger and T. Wang for technical advice and C. Heisen for critical reading of the manuscript.

Author contributions

M.I.T. performed the tissue processing, clearing and imaging experiments. M.I.T. and K.T.-V. developed the whole-brain staining protocol. M.I.T. stitched and assembled the whole-brain scans. V.E. and J.C.P. generated the synthetic vascular training dataset. J.C.P., G.T. and O.S. developed the deep learning architecture and trained the models. J.C.P. and S.S. performed the quantitative analyses. M.I.T. annotated the data. M. Düring and M. Dichgans helped with data interpretation. B.M., M.P. and G.T. provided guidance in developing the deep learning architecture and helped with data interpretation. A.E., M.I.T., B.M. and J.C.P. wrote the manuscript. All authors edited the manuscript. A.E. initiated and led all aspects of the project.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41592-020-0792-1.

Correspondence and requests for materials should be addressed to B.M. or A.E.

Peer review information Nina Vogt was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Reprints and permissions information is available at www.nature.com/reprints.

BIBLIOGRAPHY

[1] D. Ballard and J. Sklansky. "Tumor detection in radiographs." In: *Computers and Biomedical Research* 6.4 (1973), pp. 299–321. DOI: 10.1016/0010-4809(73)90066-9.

[2] D. Ballard and J. Sklansky. "A Ladder-Structured Decision Tree for Recognizing Tumors in Chest Radiographs." In: *IEEE Transactions on Computers* C-25.5 (1976), pp. 503–513. DOI: 10.1109/TC.1976.1674638.

[3] Carlos Becker, Roberto Rigamonti, Vincent Lepetit, and Pascal Fua. "Supervised Feature Learning for Curvilinear Structure Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*. Ed. by Kensaku Mori, Ichiro Sakuma, Yoshinobu Sato, Christian Barillot, and Nassir Navab. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 526–533. ISBN: 978-3-642-40811-3.

[4] Wagih Ben Hassen et al. "Inter- and intraobserver reliability for angiographic leptomeningeal collateral flow assessment by the American Society of Interventional and Therapeutic Neuroradiology/Society of Interventional Radiology (ASITN/SIR) scale." In: *Journal of NeuroInterventional Surgery* 11 (Aug. 2018), neurintsurg–2018. DOI: 10.1136/neurintsurg-2018-014185.

[5] Patrick Ferdinand Christ et al. "Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Ed. by Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells. Cham: Springer International Publishing, 2016, pp. 415–423. ISBN: 978-3-319-46723-8.

[6] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Ed. by Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells. Cham: Springer International Publishing, 2016, pp. 424–432. ISBN: 978-3-319-46723-8.

[7] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition." In: *Neural Computation* 22.12 (Dec. 2010), pp. 3207–3220. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00052. eprint: https://direct.mit.edu/neco/article-pdf/22/12/3207/842857/neco_a_00052.pdf. URL: https://doi.org/10.1162/NECO_a_00052.

[8] Dan Cireşan, Ueli Meier, and Juergen Schmidhuber. "Multi-column Deep Neural Networks for Image Classification." In: *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Feb. 2012). DOI: 10.1109/CVPR.2012.6248110.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database." In: *CVPR09*. 2009.

[10] J.S. Duncan and N. Ayache. "Medical image analysis: progress over two decades and the challenges ahead." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.1 (2000), pp. 85–106. DOI: 10.1109/34.824822.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. "The Pascal Visual Object Classes (VOC) Challenge." In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.

[12] J. Faber, W. Chilian, E. Deindl, N. van Royen, and M. Simons. "A brief etymology of the collateral circulation." In: *Arteriosclerosis, thrombosis, and vascular biology* 34 9 (2014), pp. 1854–9.

[13] Yaroslav Ganin and Victor S. Lempitsky. "N^4 -Fields: Neural Network Nearest Neighbor Fields for Image Transforms." In: *ArXiv* abs/1406.6558 (2014).

[14] Myron Ginsberg. "The cerebral collateral circulation: Relevance to pathophysiology and treatment of stroke." In: *Neuropharmacology* 134 (Aug. 2017), pp. 280–292. DOI: 10.1016/j.neuropharm.2017.08.003.

[15] Gregory Griffin, Alex Holub, and Pietro Perona. "Caltech-256 object category dataset." In: *California Institute of Technology* (2007). DOI: https://www.bibsonomy.org/bibtex/21f746f23ff0307826cca3e3be45f8de7/s364315.

[16] Jerzy W. Grzymala-Busse, Linda K. Goodwin, Witold J. Grzymala-Busse, and Xinqun Zheng. "An Approach to Imbalanced Data Sets Based on Changing Rule Strength." In: *Rough-Neural Computing: Techniques for Computing with Words*. Ed. by Sankar K. Pal, Lech Polkowski, and Andrzej Skowron. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 543–553. ISBN: 978-3-642-18859-6. DOI: 10.1007/978-3-642-18859-6_21. URL: https://doi.org/10.1007/978-3-642-18859-6_21.

[17] Haixiang Guo, Yijing Li, Jennifer S. Shang, Mingyun Gu, Huang Yuanyue, and Gong Bing. "Learning from class-imbalanced data: Review of methods and applications." In: *Expert Syst. Appl.* 73 (2017), pp. 220–239.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[19] A.D. Hoover, Valentina Kouznetsova, and Michael Goldbaum. "Locating Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response." In: *IEEE transactions on medical imaging* 19 (Apr. 2000), pp. 203–10. DOI: 10.1109/42.845178.

[20] Qing Huang, Jinfeng Sun, Hui Ding, Xiaodong Wang, and Guangzhi Wang. "Robust liver vessel extraction using 3D U-Net with variant dice loss function." In: *Computers in Biology and Medicine* 101 (2018), pp. 153–162. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2018.08.018. URL: https://www.sciencedirect.com/science/article/pii/S0010482518302385.

[21] Ivo Jansen et al. "Comparison of CTA- and DSA-Based Collateral Flow Assessment in Patients with Anterior Circulation Stroke." In: *American Journal of Neuroradiology* 37 (July 2016). DOI: 10.3174/ajnr.A4878.

[22] F. Kauw et al. "Collateral Status in Ischemic Stroke: A Comparison of Computed Tomography Angiography, Computed Tomography Perfusion, and Digital Subtraction Angiography." In: *Journal of computer assisted tomography* 44(6) (2020), pp. 984–992.

[23] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[26] Fei-Fei Li, Justin Johnson, and Serena Yeung. "Detection and Segmentation, CS231n: Convo-lutional Neural Networks for Visual Recognition." In: *Stanford University [online] [last accessed December 2022]* (2017). URL: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf.

[27] Shu Liao, Yaozong Gao, Aytekin Oto, and Dinggang Shen. "Representation Learning: A Unified Deep Learning Framework for Automatic Prostate MR Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*. Ed. by Kensaku Mori, Ichiro Sakuma, Yoshinobu Sato, Christian Barillot, and Nassir Navab. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 254–261. ISBN: 978-3-642-40763-5.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft COCO: Common Objects in Context." In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.

[29] Siqi Liu, Donghao Zhang, Yang Song, Hanchuan Peng, and Weidong Cai. "Triple-Crossing 2.5D Convolutional Neural Network for Detecting Neuronal Arbours in 3D Microscopic Images." In: *Machine Learning in Medical Imaging*. Ed. by Qian Wang, Yinghuan Shi, Heung-Il Suk, and Kenji Suzuki. Cham: Springer International Publishing, 2017, pp. 185–193. ISBN: 978-3-319-67389-9.

[30] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. "Deep Retinal Image Understanding." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Ed. by Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells. Cham: Springer International Publishing, 2016, pp. 140–148. ISBN: 978-3-319-46723-8.

[31] Ueli Meier, Dan Ciresan, Luca Maria Gambardella, and Juergen Schmidhuber. "Better Digit Recognition with a Committee of Simple Neural Nets." In: *2011 International Conference on Document Analysis and Recognition*. Oct. 2011, pp. 1250 –1254. DOI: 10.1109/ICDAR.2011.252.

[32] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation." In: *2016 Fourth International Conference on 3D Vision (3DV)* (2016), pp. 565–571.

[33] I. Muehlen, S. P. Kloska, P. Gölitz, P. Hölter, L. Breuer, H. Ditt, and A. Doerfler. "Noninvasive Collateral Flow Velocity Imaging in Acute Ischemic Stroke: Intraindividual Comparison of 4D-CT Angiography with Digital Subtraction Angiography." In: *Rofo* (2019), pp. 827–835.

[34] José Ignacio Orlando and Matthew Blaschko. "Learning Fully-Connected CRFs for Blood Vessel Segmentation in Retinal Images." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*. Ed. by Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe. Cham: Springer International Publishing, 2014, pp. 634–641. ISBN: 978-3-319-10404-1.

[35] S. M. Pizer and A. E. Todd-Pokropek. "Improvement of scintigrams by computer processing." In: *Seminars in nuclear medicine* 8.2 (1978), pp. 125–146. DOI: 10.1016/s0001-2998(78)80036-1.

[36] Adhish Prasoon, Kersten Petersen, Christian Igel, François Lauze, Erik Dam, and Mads Nielsen. "Deep Feature Learning for Knee Cartilage Segmentation Using a Triplanar Convolutional Neural Network." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*. Ed. by Kensaku Mori, Ichiro Sakuma, Yoshinobu Sato, Christian Barillot, and Nassir Navab. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 246–253. ISBN: 978-3-642-40763-5.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[38] Holger R. Roth, Le Lu, Ari Seff, Kevin M. Cherry, Joanne Hoffman, Shijun Wang, Jiamin Liu, Evrim Turkbey, and Ronald M. Summers. "A New 2.5D Representation for Lymph Node Detection Using Random Sets of Deep Convolutional Neural Network Observations." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*. Ed. by Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe. Cham: Springer International Publishing, 2014, pp. 520–527. ISBN: 978-3-319-10404-1.

[39] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. "LabelMe: A Database and Web-Based Tool for Image Annotation." In: *International Journal of Computer Vision* 77 (2007), pp. 157–173.

[40] Matthias Schneider, Sven Hirsch, Bruno Weber, Gábor Székely, and Bjoern H. Menze. "TGIF: Topological Gap In-Fill for Vascular Networks." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*. Ed. by Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe. Cham: Springer International Publishing, 2014, pp. 89–96. ISBN: 978-3-319-10470-6.

[41] Matthias Schneider, Johannes Reichold, Bruno Weber, Gábor Székely, and Sven Hirsch. "Tissue metabolism driven arterial tree generation." In: *Medical Image Analysis* 16.7 (2012). Special Issue on the 2011 Conference on Medical Image Computing and Computer Assisted Intervention, pp. 1397–1414. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2012.04.009. URL: https://www.sciencedirect.com/science/article/pii/S1361841512000576.

[42] Anjany Sekuboyina, Markus Rempfler, Jan Kukačka, Giles Tetteh, Alexander Valentinitsch, Jan S. Kirschke, and Bjoern H. Menze. "Btrfly Net: Vertebrae Labelling with Energy-Based Adversarial Learning of Local Spine Prior." In: *Medical Image Computing and Computer Assisted Intervention –*

MICCAI 2018. Ed. by Alejandro F. Frangi, Julia A. Schnabel, Christos Davatzikos, Carlos Alberola-López, and Gabor Fichtinger. Cham: Springer International Publishing, 2018, pp. 649–657.

[43] S. A. Sheth and D. S. Liebeskind. "Imaging Evaluation of Collaterals in the Brain: Physiology and Clinical Translation." In: *Current radiology reports* 29 (2014). DOI: https://doi.org/10.1007/s40134-013-0029-5.

[44] Nahian Siddique, Sidike Paheding, Colin P. Elkin, and Vijay Devabhaktuni. "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications." In: *IEEE Access* 9 (2021), pp. 82031–82057. DOI: 10.1109/ACCESS.2021.3086020.

[45] Patrice Y. Simard, David Steinkraus, and John C. Platt. "Best practices for convolutional neural networks applied to visual document analysis." In: *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* (2003), pp. 958–963.

[46] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1409.1556.

[47] Joes Staal, Michael Abràmoff, Meindert Niemeijer, Max Viergever, and Bram Ginneken. "Ridge-Based Vessel Segmentation in Color Images of the Retina." In: *IEEE transactions on medical imaging* 23 (Apr. 2004), pp. 501–9. DOI: 10.1109/TMI.2004.825627.

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

[49] Giles Tetteh, Velizar Efremov, Nils D. Forkert, Matthias Schneider, Jan Kirschke, Bruno Weber, Claus Zimmer, Marie Piraud, and Björn H. Menze. "DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes." In: *Frontiers in Neuroscience* 14 (2020). ISSN: 1662-453X. DOI: 10.3389/

fnins.2020.592352. URL: https://www.frontiersin.org/articles/10.3389/fnins.2020.592352.

[50] Giles Tetteh, Fernando Navarro, Raphael Meier, Johannes Kaesmacher, Johannes C. Paetzold, Jan S. Kirschke, Claus Zimmer, Roland Wiest, and Bjoern H. Menze. "A deep learning approach to predict collateral flow in stroke patients using radiomic features from perfusion images." In: *Frontiers in Neurology* 14 (2023). ISSN: 1664-2295. DOI: 10.3389/fneur.2023.1039693. URL: https://www.frontiersin.org/articles/10.3389/fneur.2023.1039693.

[51] Giles Tetteh, Markus Rempfler, Claus Zimmer, and Bjoern H. Menze. "Deep-FExt: Deep Feature Extraction for Vessel Segmentation and Centerline Prediction." In: *Machine Learning in Medical Imaging*. Ed. by Qian Wang, Yinghuan Shi, Heung-Il Suk, and Kenji Suzuki. Cham: Springer International Publishing, 2017, pp. 344–352. ISBN: 978-3-319-67389-9.

[52] Mihail Todorov et al. "Machine learning analysis of whole mouse brain vasculature." In: *Nature Methods* 17 (Apr. 2020), pp. 1–8. DOI: 10.1038/s41592-020-0792-1.

[53] Antonio Torralba, Rob Fergus, and William T. Freeman. "80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2008), pp. 1958–1970.

[54] Jianxiong Xiao, James Hays, Krista Ehinger, Aude Oliva, and Antonio Torralba. "SUN database: Large-scale scene recognition from abbey to zoo." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. June 2010, pp. 3485–3492. DOI: 10.1109/CVPR.2010.5539970.

[55] Saining Xie and Zhuowen Tu. "Holistically-Nested Edge Detection." In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1395–1403. DOI: 10.1109/ICCV.2015.164.

[56] Lina Xu, Giles Tetteh, Jana Lipkova, Yu Zhao, Hongwei Li, Patrick Christ, Marie Piraud, Andreas Buck, Kuangyu Shi, and Bjoern H. Menze. "Automated Whole-Body Bone Lesion Detection for Multiple Myeloma on 68Ga-Pentixafor PET/CT Imaging Using Deep Learning Methods." In: *Con-*

trast Media & Molecular Imaging 2018 (2018). DOI: 10.1155/2018/2391925.

[57] Lina Xu, Giles Tetteh, Mona Mustafa, Jana Lipkova, Yu Zhao, Marie Bieth, Patrick Christ, Marie Piraud, Bjoern Menze, and Kuangyu Shi. "W-Net for Whole-Body Bone Lesion Detection on 68Ga-Pentixafor PET/CT Imaging of Multiple Myeloma Patients." In: *Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment*. Cham: Springer International Publishing, 2017, pp. 23–30. ISBN: 978-3-319-67564-0.

[58] Guodong Zeng, Xin Yang, Jing Li, Lequan Yu, Pheng-Ann Heng, and Guoyan Zheng. "3D U-net with Multi-level Deep Supervision: Fully Automatic Segmentation of Proximal Femur in 3D MR Images." In: *Machine Learning in Medical Imaging*. Ed. by Qian Wang, Yinghuan Shi, Heung-Il Suk, and Kenji Suzuki. Cham: Springer International Publishing, 2017, pp. 274–282. ISBN: 978-3-319-67389-9.

[59] Chen Zhao, Jungang Han, Yang Jia, and Fan Gou. "Lung Nodule Detection via 3D U-Net and Contextual Convolutional Neural Network." In: *2018 International Conference on Networking and Network Applications (NaNA)*. Oct. 2018, pp. 356–361. DOI: 10.1109/NANA.2018.8648753.