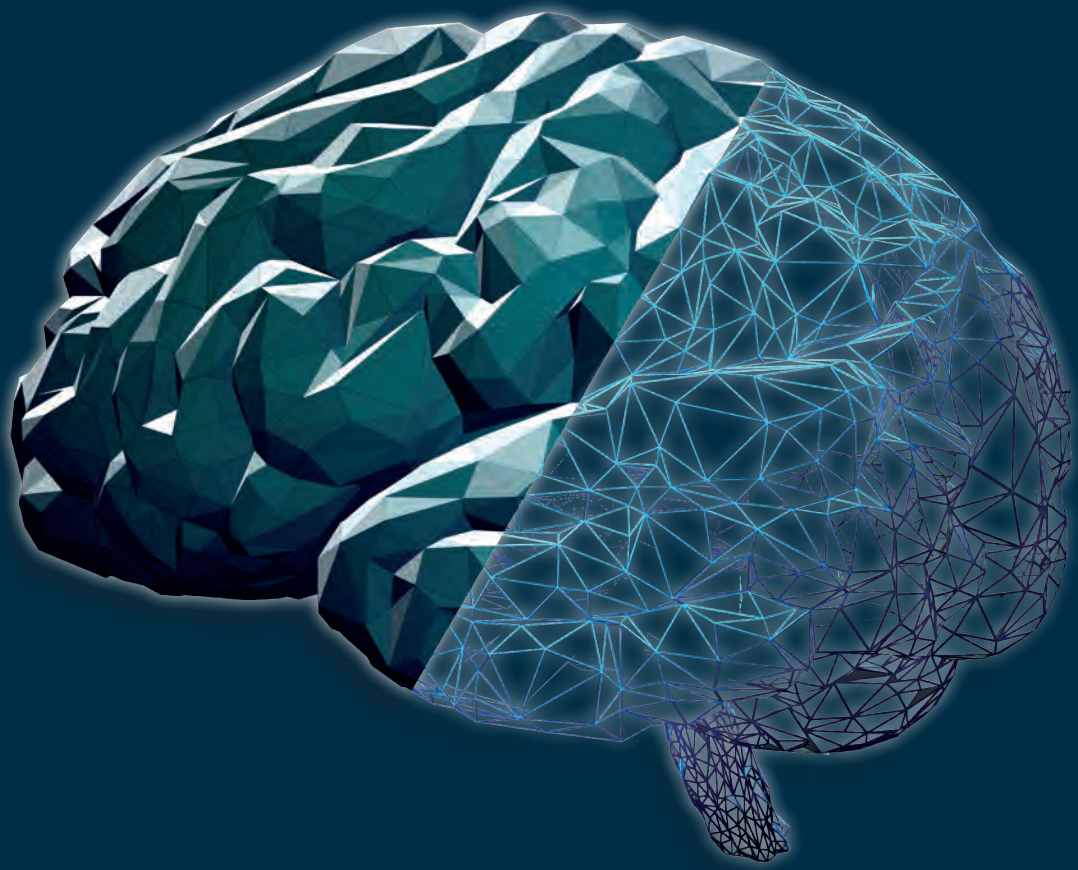
Efficient neural network training for 4D-CTA stroke imaging



Sil van de Leemput

RADBOUD UNIVERSITY PRESS

Radboud Dissertation Series

Efficient neural network training for 4D-CTA stroke imaging

Silvester Christiaan van de Leemput

The research described in this thesis was carried out at the Diagnostic Image Analysis Group, Radboud University Medical Center (Nijmegen, the Netherlands) within the Radboud Institute for Health Sciences.
This work was funded by The Netherlands Organization for Scientific Research (NWO) and Canon Medical Systems Corporation, Japan.

Author: Sil van de Leemput
Title: Efficient neural network training for 4D-CTA stroke imaging

Radboud Dissertations Series
ISSN: 2950-2772 (Online); 2950-2780 (Print)

Published by RADBOUD UNIVERSITY PRESS
Postbus 9100, 6500 HA Nijmegen, The Netherlands
www.radbouduniversitypress.nl

Design: Sil van de Leemput
Cover: Proefschrift AIO | Guntra Laivacuma and Sil van de Leemput
Printing: DPN Rikken/Pumbo

ISBN: 978-94-93296-63-3
DOI: 10.54195/9789493296633
Free download at:
www.boekenbestellen.nl/radboud-university-press/dissertations

RADBOUD
UNIVERSITY
PRESS

Efficient neural network training for 4D-CTA stroke imaging

Proefschrift

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. dr. J.M. Sanders,
volgens besluit van het college voor promoties
in het openbaar te verdedigen op

dinsdag 10 september 2024
om 12.30 uur precies

door

Silvester Christiaan van de Leemput

geboren op 19 maart 1987
te Haarlemmermeer

Promotoren
Prof. dr. B. van Ginneken
Prof. dr. W.M. Prokop

Copromotor
Dr. ir. R. Manniesing

Manuscriptcommissie
Prof. dr. H.F. de Leeuw
Prof. dr. A. Vilanova *(Technische Universiteit Eindhoven)*
Dr. E. Gavves *(Universiteit van Amsterdam)*

Efficient neural network training for 4D-CTA stroke imaging

Dissertation

to obtain the degree of doctor
from Radboud University Nijmegen
on the authority of the Rector Magnificus prof. dr. J.M. Sanders,
according to the decision of the Doctorate Board
to be defended in public on

Tuesday, September 10, 2024
at 12.30 pm

by

Silvester Christiaan van de Leemput

born on March 19, 1987
in Haarlemmermeer (the Netherlands)

PhD supervisors
Prof. dr. B. van Ginneken
Prof. dr. W.M. Prokop

PhD co-supervisors
Dr. ir. R. Manniesing

Manuscript Committee
Prof. dr. H.F. de Leeuw
Prof. dr. A. Vilanova *(Technical University Eindhoven)*
Dr. E. Gavves *(University of Amsterdam)*

Table of contents

1 Introduction **1**
- 1.1 Acute stroke 3
- 1.2 Computed tomography 4
- 1.3 Acute stroke workup 5
- 1.4 Machine learning 7
- 1.5 Outline of this thesis 11

2 Multiclass brain tissue segmentation in 4D-CTA **15**
- 2.1 Introduction 17
- 2.2 Methods 19
- 2.3 Data 24
- 2.4 Experiments 27
- 2.5 Results 30
- 2.6 Discussion 32

3 Deriving 3D non-contrast CT from 4D-CTA **41**
- 3.1 Introduction 43
- 3.2 Methods 46
- 3.3 Data 50
- 3.4 Evaluation 51
- 3.5 Experiments 53
- 3.6 Results 55
- 3.7 Discussion 60

4 MemCNN: a framework for memory efficient invertible networks **67**
- 4.1 Introduction 69
- 4.2 Methods 70
- 4.3 Experiments and results 74
- 4.4 Works using MemCNN 75
- 4.5 Conclusion 75

5 Discussion **79**
- 5.1 4D-CTA for acute stroke imaging 81
- 5.2 Deep learning for acute stroke imaging 83
- 5.3 Beyond acute stroke imaging 83
- 5.4 Future research 86
- 5.5 Concluding remarks 88

Summary **93**

Samenvatting **97**

Publications **101**

Data management **103**

PhD portfolio **105**

Bibliography **107**

Dankwoord **119**

Curriculum vitae **123**

1

Introduction

1.1 Acute stroke

Stroke is a disturbance of blood flow in the brain leading to cell-death. Stroke events typically occur suddenly and require fast medical attention. According to the World Stroke Organization, stroke is the second leading cause of death and is also the third leading cause of disability worldwide[1]. Common stroke symptoms are sudden numbness or inability to move the arm or leg, confusion, slurred speech, and loss of vision, which varies by severity and location of the stroke. Approximately, one out of four adults over the age of 25 will have a stroke in their lifetime, furthermore, 12.2 million people are having a stroke for the first time and 6.5 million will die as a result of stroke each year[1].

There are two major different stroke types: hemorrhagic stroke and ischemic stroke. Hemorrhagic stroke refers to the rupture of blood vessels in the brain, which comprises approximately 13% of all stroke cases[2]. This may occur either within the brain (intra-cranial hemorrhage) or below the dura (subarachnoid hemorrhage). Ischemic stroke, on the other hand, is caused by the blockage of blood vessels by a thrombus, usually in the form of a blood clot. This is the most common form of acute stroke, accounting for approximately 87% of all stroke cases[2]. A simplified overview of the two major types of stroke can be seen in Figure 1.1.

As many as 1.9 million brain cells die for every minute the brain is deprived of blood[3]. Therefore, timely treatment is essential to have a good patient outcome, and hence the phrase *time is brain* is often used with respect to treating acute stroke. In the case of an ischemic stroke, the thrombus or clot should be removed to restore blood flow. This can be achieved by injecting clot-resolving drugs into the arm or administering them directly inside the blocked blood vessel using a thin tube through an artery in the groin. Alternatively, a surgical procedure called a (mechanical) thrombectomy can be applied. Here, the surgeon uses a device attached to a catheter to directly remove blood clots. Treatment of hemorrhagic stroke is much more difficult and focuses on controlling the bleeding and reducing the pressure in the brain caused by excess fluid. Treatment options include the administration of drugs, surgery to remove blood and relieve pressure on the brain, or even surgery to repair vessels or stop the source of the bleeding. The key point to make here is that a fast and accurate diagnosis of stroke symptoms is essential for preventing death and for achieving the best possible patient outcome.

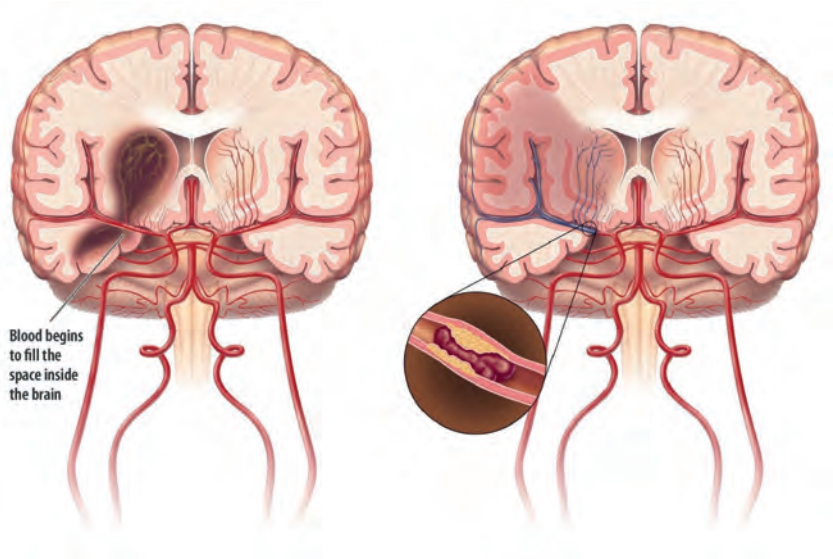


Figure 1.1: Simplified examples of the two major types of stroke. On the left hemorrhagic stroke where a ruptured bloodvessel leads to intracerebral bleeding. On the right an ischemic stroke where an occluded blood vessel deprives a large region of brain tissue of oxygen. Adapted from: https://www.strokecenter.org/ (January 2020).

1.2 Computed tomography

Computed Tomography (CT) is the foremost diagnostic imaging tool for stroke imaging[4]. It relies on measuring the difference in tissue density using penetrating X-rays. X-rays were discovered by Wilhelm Conrad Röntgen and are a form of electromagnetic radiation[5]. Conventional X-ray imaging uses an X-ray tube to generate radiation that is subsequently detected by a static detector. Dense materials (e.g, bones) absorb X-rays to a higher degree than soft materials (e.g., muscles), which allows the creation of a 2D image that shows the difference in material density for the scanned materials or tissues. Tissue density is measured in Hounsfield Units (HU). For example, water has a value of 0 HU, and air has a value of -1000 HU, whereas bone can have a value in the range of 300-1900 HU.

Modern CT scanners, as shown in Figure 1.2, are no longer reliant on a static X-ray detector, but instead can take multiple images from various angles of a subject that can be combined in a volumetric (3D) reconstruction. This feature is achieved by rotating the X-ray tube and opposing X-ray detector in a circular gantry around the scanned subject while taking multiple 2D images at various angles. Using a

mathematical reconstruction algorithm on the 2D images acquired this way results in a 3D volumetric representation of the scanned subject. Modern CT scanners are now even able to acquire several 3D volumetric acquisitions in quick succession, resulting in a 4D acquisition that can be used to examine dynamic phenomena like blood flow.

Figure 1.2: Computed tomography scanner. An X-ray tube and an oppositely placed detector are situated within the circular gantry at the head of the scanner bench. The detector measures the X-ray beam attenuation of a subject on the bench creating a projection image. The gantry internals can rotate at approximately two to three revolutions per second. Image source: https://us.medical.canon/ (September 2022). Copyright: Canon Medical Systems.

1.3 Acute stroke workup

As we explained above, the acute nature of stroke requires a fast diagnosis from medical specialists. The main diagnostic imaging tool for stroke imaging is therefore Computed Tomography (CT). CT exams can be performed in a matter of seconds to minutes. The first diagnostic priority is to differentiate between hemorrhagic and ischemic stroke. CT scans can be made just after the administration of contrast agent, which is a substance intravenously injected that can cause blood to be seen more clearly. The difference between a contrast-enhanced and non-contrast CT image is shown in Figure 1.3. A non-contrast CT (NCCT) scan allows the exclusion of intracerebral hemorrhage and lesions that might mimic acute ischemic stroke such as

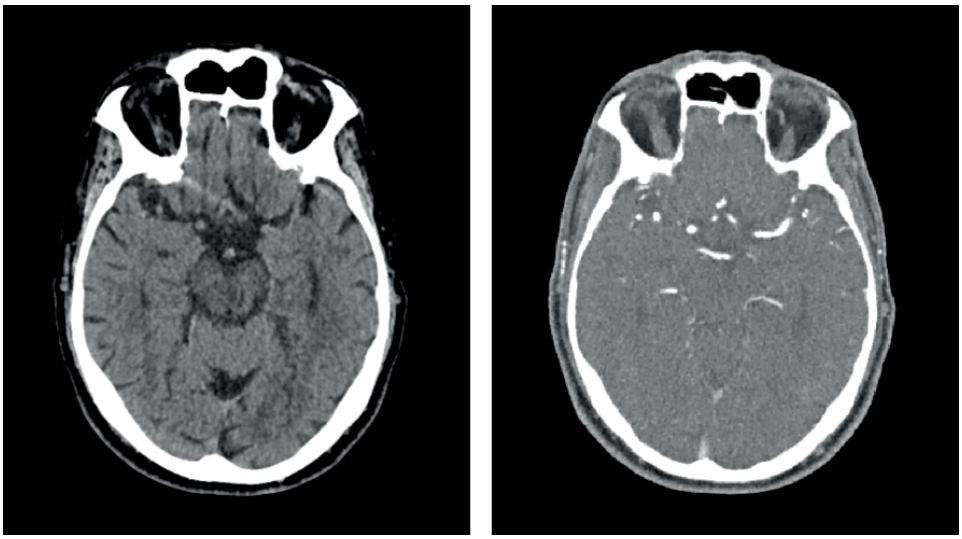


Figure 1.3: Example of axial cross-sections of the head of a non-contrast CT (NCCT) image on the left and a CT angiography (CTA) image with contrast on the right.

tumor or intracerebral hemorrhage when using a contrast scan instead. Therefore, a non-contrast CT (NCCT) scan is typically performed first.

When there is no hemorrhagic stroke visible on the NCCT image, CT angiography (CTA) and/or 4D-CTA angiography (4D-CTA) scans are taken to identify potential blood-deprived areas in the brain and their causes. Both CTA and 4D-CTA scans are taken after injection of contrast agent. Once the contrast agent travels through the vasculature of the brain, the scan is made, making the vessels light up bright on the CT image. The images can be used to look for abnormalities in the vasculature, like the lack of blood flow due to occlusions or stenosis.

The major difference between a CTA and a 4D-CTA scan is that whereas the former only uses a single 3D acquisition of the contrast agent in the brain, the latter takes multiple 3D acquisitions over time resulting in a 4D acquisition of the brain in which the flow of contrast agent in the cerebral vasculature is captured. The added dynamic information contained in the 4D-CTA acquisition is conventionally used to compute perfusion maps (e.g., cerebral blood flow, cerebral blood volume, and mean transit time), to detect perfusion defects, and for the estimation of infarct core and penumbra region. Here the infarct core is irreversibly damaged tissue and the penumbra is the tissue that is still salvageable once the blood supply is restored. The 4D-CTA is a rich and challenging source of data for ischemic stroke diagnosis. Because modern CT scanners generate data with a higher spatial and temporal resolution, and the abnormalities to look for are often small and subtle, the image in-

terpretation process, carried out by neuroradiologists, is becoming increasingly time-consuming and tedious. Hence, machine learning methods that can automate and support diagnosis are becoming increasingly relevant.

1.4 Machine learning

Machine learning is a field of artificial intelligence that aims to build learning algorithms that generate programs that accurately perform a task without being explicitly programmed to do so. A learning algorithm can be understood using the following definition by Mitchell[6]: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." This definition encapsulates the key components of machine learning algorithms, where a wide variety of experiences E, tasks T, and performance measures P can be picked. Ultimately, a machine learning algorithm produces a model or algorithm from experience E that can be expressed as a function $y(x)$, which takes a task-related input x and produces a task-related output y, which should perform well (on performance metric P) on task T even for new unseen experiences not within E.

Some of the most common categories of machine learning tasks T are classification (e.g., finding a categorical/discreet label for a given input), regression (e.g., assigning a continuous value for a given input), and image segmentation (e.g., finding categorical/discreet labels for each pixel in an input image). Furthermore, machine learning generally focuses on tasks T that are too difficult or too time-consuming to solve with rule-based programs designed and implemented by humans (essentially where humans program the function $y(x)$ themselves). For example, in chapter 2 we segment different brain tissues in 4D-CTA images into white matter, gray matter, cerebrospinal fluid, and blood vessels, by applying a machine learning method. We could attempt to manually segment these tissues by directly applying intensity-based thresholds on the input image. However, this would not exactly coincide with the exact tissue types, due to overlap and ambiguity in the HU for certain tissue types and noise in the 4D-CTA images. To improve upon this initial result we would have to iteratively design new rules or preprocessing steps and test if these improve the system. Yet, with a machine learning approach, we can attempt to learn many of these rules directly from the data, saving time and ultimately discovering rules that we might not have even thought of that result in better segmentation performance.

Learning algorithms can be roughly categorized as being supervised or unsupervised, which changes the available experience E (and usually also the used performance metrics P) during the learning process. This thesis will solely focus on super-

vised learning algorithms which is an experience E with labeled examples. That is, we are provided with a training set of given inputs x for which we also have the corresponding expected target labels $y(x)$ available. This is opposed to unsupervised learning algorithms, which have experiences E with only the inputs x available. The goal in the unsupervised setting is usually to find patterns in the input data x, such as compact representations or clusters.

As a concrete example of a supervised machine learning algorithm, we will take the regression task T from Chapter 3 where we attempt to reconstruct a NCCT image from a 4D-CTA image. The input of the algorithm $y(x)$ is a 4D-CTA image x represented as a vector of voxels and the target output y is represented as a vector of voxels of a NCCT image that is registered to the associated 4D-CTA image. As experience E we use a training dataset of a number of corresponding 4D-CTA and NCCT pairs that are properly spatially aligned. As performance metric P we take the mean squared error, which estimates the voxel-wise difference between the reconstructed image from the algorithm NCCT* and the reference NCCT y. Ultimately the goal is to show that the 'learned' model $y(x)$ performs well on examples that are not contained in the training dataset from experience E. This concept is called how well a model 'generalizes' and for this purpose, so we used a separate testing dataset to test the generalization capability of the learned model.

1.4.1 Applications

Machine learning algorithms are used all around us in everyday life, although most of them often appear invisible to most people since they don't require active interaction or awareness of the user. Google uses the search queries entered by their users to learn to improve their services, companies like Facebook, Amazon, and Netflix will learn about user preferences while users are interacting with their systems, supermarkets find spending patterns by analyzing customer buying behavior, banks use fraud detection algorithms on financial transactions, and smartphones and smartwatches have algorithms to predict and monitor user activity. Algorithms that are more visible to people are voice recognition and synthesis software, image and video filters for object and face recognition, self-driving cars, and various modern and popular content creation tools that are able to generate high-fidelity output based on simple text prompt inputs like DALL-E for image generation and ChatGPT for text, code, and documentation synthesis. All machine learning algorithms generally have model parameters that are tuned for the task they are designed to solve.

Usually, there is a close relation between the number of required model parameters and the size of an algorithm input x. A model for a simple task like learning to

predict user preferences from a few variables such as age, sex, and demographic information would likely require substantially fewer model parameters than for example predicting whether a medical image scan contains a certain disease or if a raw audio signal contains a certain utterance. When the input data is complex and a model thus requires more parameters, this complicates optimizing or training the machine learning model. It may be more computationally expensive and thus time-consuming to find the optimal set of parameters and the model may need more annotated data to optimize all of its parameters accurately. Hence, traditional machine learning approaches have often tried to circumvent this problem to extract task-related features to reduce the complexity of the input signal.

Feature extraction is a core concept within machine learning, where task-specific information is either extracted from the input data x or added a-priori in relation to task T. Traditionally, these features were hand-picked and implemented by human programmers. For example, in chapter 2 we derived two lower-dimensional feature maps, called the weighted temporal average (WTA) and the weighted temporal variance (WTV), from the high-dimensional 4D-CTA images which we used as input data for a brain tissue segmentation task. Generally, well-chosen task-specific features can help to increase task performance and can reduce the overall time required to optimize the machine learning model. However, feature extraction has the drawback that it introduces a bias toward how humans think the task needs to be solved. Also, it reduces the generalizability of the machine learning algorithm, since it can introduce features that might not translate well to other tasks. More modern approaches like deep learning, which is the predominant approach throughout this thesis, tend to move away from human hand-crafted feature extraction to a more generic and data-centric approach and models that can be learned end-to-end from the data.

1.4.2 Deep Learning

Deep learning is a sub-field of machine learning that focuses primarily on artificial neural networks (ANNs) consisting of a structured network of multiple small simple interconnected computational units as a machine learning model. Deep neural networks are obtained by stacking multiple configurations of these ANN models — hence the word deep — in such a way it allows them to, almost magically, perform increasingly complex tasks. Although the idea for deep learning was already around since the late seventies[7,8], it took several decades to have sufficiently large annotated datasets, sufficient computational power, and efficient training methods to make these models work.

The availability of large annotated datasets has had a big impact on the develop-

ment of deep learning as a field. For example, the ImageNet[9] challenge used a big database of nearly 15 million manually labeled natural images, that were organized into 21 thousand separate classes. Especially for more complex deep learning models with an extensive number of parameters, the availability of these datasets helps to supply sufficient examples to better estimate the optimal value for these model parameters. Having these curated publicly available reference datasets also makes the comparisons of different approaches easier by referencing the used dataset and the task to solve. Within hospitals, digitized patient data like CT scans are increasingly collected. Although curated annotated data are often still scarce, these are increasing as well. Altogether the availability of digitized data helps to increase the development of deep learning approaches for automating several tasks within the medical domain, like the works in chapters 2 and 3 that rely on NCCT and 4D-CTA scans from our hospital.

The current backbone of modern deep learning weight optimization (the actual learning of the model parameters from the data) is based on the backpropagation algorithm in combination with the stochastic gradient descent algorithm. The backpropagation algorithm was generalized for neural networks by Rumelhart et al.[10], which efficiently computes the gradients of the objective function, with respect to the parameter weights for deep neural networks, by dynamically traversing the gradient from the loss function from the last layer back to the first layer using the chain-rule. The backpropagation algorithm does not deal with how the gradients are used to update the weights, which is instead typically done by using a form of the gradient descent algorithm[11], which is an iterative method to estimate for each of the weights in which direction to update them (either to increase or to decrease) in order to optimize the objective function given the data from the training dataset.

Deep learning models require large datasets to achieve good performance. However, the size of the models can become a burden on the hardware used to train the models. Traditionally, neural networks were trained by adjusting the parameters that work best on average given the whole training dataset, i.e., using the gradient descent algorithm. This might work for small datasets, but for example, ImageNet is approximately 150GB in size, which for a typical desktop computer doesn't fit all at once in memory. Hence, a modified version of the gradient descent algorithm was invented called stochastic gradient descent (SGD)[12,13], which does not adjust the parameters given the whole training dataset, but instead does so on smaller chunks of the training data called mini-batches that are processed independently.

Another key success factor for developing and training deep neural networks is the increasing computational power available. Deep learning model training is computationally expensive in the number of computational operations and typically re-

quires specialized hardware to perform in reasonable time like graphic processor units (GPUs) which were originally designed for gaming or even Google's tensor processing units (TPUs) which were specifically designed for deep learning training and inference. The specialized hardware is designed to take advantage of the fact that most of the small individual units within the network can be independently computed, which allows for parallel computations to reduce overall computation time.

The reliance on specialized hardware like GPUs and TPUs introduces new problems for researchers and the industry. For example, during deep learning model training with sufficiently large input data for a learning task it is possible that the required memory to compute all the intermediate states required for the backpropagation algorithm on the specialized hardware is insufficient even for small batch sizes. These cases can often be solved using some workarounds, like offloading some of the intermediate results to disk or normal RAM, using checkpoints, and splitting the model over multiple GPUs (model parallelism). Ultimately these workarounds will result in concessions regarding model complexity, training efficiency, and training time, and on top of that often cost a lot of additional effort for the programmer. To supply the deep learning developers with tools to reduce memory requirements during training we have developed a PyTorch framework called MemCNN in chapter 4.

1.5 Outline of this thesis

4D-CTA data is a useful imaging modality for assessing stroke symptoms but is also challenging due to the high dimensionality of the data (3D + time). This thesis aims to contribute to the analysis of 4D-CTA data for stroke applications using deep learning techniques and is part of a larger research project together with Ajay Patel, and Midas Meijs and was supervised by Bram van Ginneken and Rashindra Manniesing. The research of Midas Meijs was predominantly focused on the analysis of cerebral vasculature and vascular pathology like ischemic stroke[14–17]. Whereas, the research of Ajay Patel focused on segmentation of the fundamental cerebral structures like the cranial cavity and the two hemispheres and also hemorrhagic stroke identification and quantification[18–21]. The work in this thesis focuses on the extraction of useful stroke-related information directly from the 4D-CTA data and also on deep learning techniques to reduce memory overhead during neural network training.

More specifically the rest of the content of the chapters in this thesis are as follows:

Chapter 2 describes a deep learning method to automatically segment and label brain tissue in 4D-CTA data into white matter, gray matter, cerebrospinal fluid, and vasculature.

Chapter 3 introduces a deep learning method to reconstruct a 3D non-contrast CT from 4D-CTA data, which could potentially simplify the stroke workup.

Chapter 4 presents a deep learning framework aimed at introducing support for memory optimization during model training using reversible operations.

Chapter 5 gives a general discussion of the presented work in this thesis in relation to the research field and provides suggestions for future research.

2

Multiclass brain tissue segmentation in 4D-CTA

S.C. van de Leemput, M. Meijs, A. Patel, F.J.A. Meijer, B. van Ginneken, R. Manniesing

Original title: Multiclass Brain Tissue Segmentation in 4D CT using Convolutional Neural Networks

Published in: IEEE Access, 7:51557-51569, 2019

Abstract

4D-CTA imaging has great potential for use in stroke workup. A fully convolutional neural network (CNN) for 3D multiclass segmentation in 4D-CTA is presented, which can be trained end-to-end from sparse 2D annotations. The CNN was trained and validated on 42 4D-CTA acquisitions of the brain of patients with suspicion of acute ischemic stroke. White matter, gray matter, cerebrospinal fluid, and vessels were annotated by two trained observers. The mean Dice coefficients, contour mean distances, and absolute volume differences were respectively 0.87 ± 0.04, 0.52 ± 0.47 mm, and 11.78 ± 9.55 % on a separate test set of five patients, which were similar to the average interobserver variability scores of 0.88 ± 0.03, 0.72 ± 0.93 mm, and 8.86 ± 7.65 % outperforming the current state-of-the-art. The proposed method is therefore a promising deep neural network for multiclass segmentation in 4D spatiotemporal imaging data.

2.1 Introduction

Computed tomography (CT) is at the core of modern acute stroke workup[22]. CT is cheap, widely available, and fast compared to other imaging modalities like magnetic resonance imaging (MRI). Additionally, modern CT scanners can cover the whole brain with high temporal and spatial resolution. From a head CT scan tissue densities can be derived, which enables detecting pathology like hemorrhages. Additionally, acquiring a head CT shortly after injection of contrast agent enables the visualization of the cerebral vasculature and hemodynamics. CT angiography (CTA) and 4D CT angiography (4D-CTA) are two such post-contrast techniques, which are respectively a single 3D CT scan and a series of 3D CT scans over time. This work focuses on the latter type of acquisition since we expect 4D-CTA to be the future image modality for stroke. Essentially, 4D-CTA contains more temporal information and the CTA can be derived from the 4D-CTA by a maximum intensity projection[23].

4D-CTA imaging will become increasingly important in the clinical workup of acute stroke. It can be used to assess penumbra, infarct core, and collateral flow, which can be used for selecting stroke patients for reperfusion therapy[24]. A recent prospective clinical trial showed that 4D-CTA imaging helps in identifying patients who will benefit from endovascular treatment beyond the recommended time window of six hours[25]. Segmentation of soft tissue is important because it enables tissue-dependent perfusion analysis, potentially refining the identification of infarction core and penumbra[26]. Segmentation of the cerebral vasculature is important for many applications[22,27,28]. We have demonstrated that it can be used to visualize vascular flow disturbances reducing the time to detect abnormalities such as vascular occlusion and arteriovenous malformations[29]. Despite the potential uses of 4D-CTA imaging for stroke, little work has been done on automatic segmentation of tissues from 4D-CTA data using computer algorithms.

Only one related method was found for 4D-CTA[30] which was based on a traditional pattern recognition approach. Although Manniesing et al.[30] does provide a coarse segmentation for cerebrospinal fluid (CSF) and vessels, the quantitative evaluation was only done for white matter (WM) and gray matter (GM), and only in the axial direction of slices at specific brain locations. To our knowledge, a full multiclass 3D segmentation method that includes WM, GM, CSF, and vessels in 4D-CTA and that has been quantitatively evaluated for all classes, is currently nonexistent.

In this work, we present a method for 3D multiclass segmentation in 4D-CTA using a multiresolution fully convolutional neural network (CNN) which is able to learn end-to-end from 2D sparse annotations. The CNN is applied to 4D-CTA images of acute ischemic stroke patients for segmentation of WM, GM, CSF, and vessels.

Medical imaging has witnessed a sharp rise of applications based on convolutional neural networks (CNNs) in a few years time[31]. CNNs are feed-forward artificial neural networks consisting of multiple convolutional layers successively encoding higher abstract representations. A powerful trait of CNNs is that representations can be directly learned from data without the need for manually creating or selecting features.

However, many deep learning approaches avoid learning from high dimensional data because of practical limitations, i.e., higher GPU memory requirements and increased number of computations. For example,[32–38] propose a 2.5D approach in which multiple 2D patches are sampled in different orientations around a center voxel in 3D, and are then fed individually into a 2D CNN for predicting the output class at the intersection. This approach is suboptimal since 3D context outside of the sampled planes is ignored.

Full 3D approaches have been proposed to a lesser extent. Most provide fully convolutional approaches that include multiresolution contextual information by processing downscaled versions of the input and integrating the lower resolution images later in the network at the original voxel resolution[39–44]. 3D U-Net[40] is a fully convolutional network that processes 3D input at four different image resolutions and provides a voxel weighting scheme and smooth deformation field data augmentation to be able to learn from sparsely annotated data. Other 3D segmentation approaches try to leverage recurrent operations[45–48]. Some segmentation approaches[49] utilize CNNs for processing multi-channel 3D data, but the channels represent data from different modalities, whereas 4D spatiotemporal data represents multiple acquisitions using the same modality over time. The distinction is useful, since the voxel intensities encode for similar physical phenomena in the latter case, hence calculating statistics (e.g., averages, variance) over the temporal dimension becomes sensible. For example, consider carefully registered temporal images, taking a temporal average yields a meaningful image, since its voxel intensities are approximately similar. However, for multi-model data, for example MR T0, T1, and Flair images, averaging over its channels is less meaningful since the voxel intensities do not correspond between channels. Only a single work was found in the literature that addressed 4D spatiotemporal data[50] for automatic multi-organ detection in MR using unsupervised deep learning techniques. However, the resulting segmentations have limited precision and class overlap.

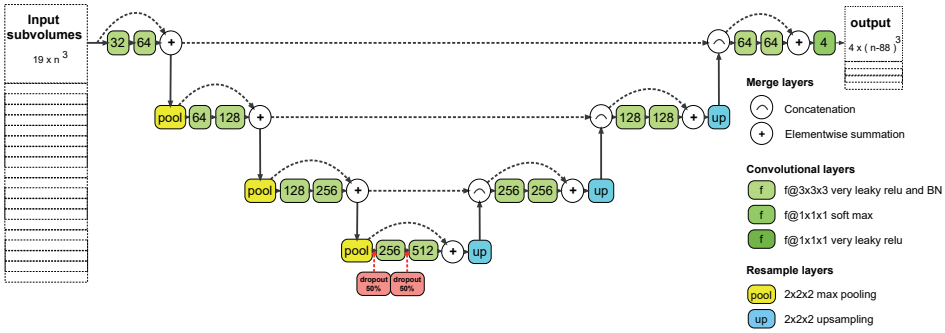


Figure 2.1: Our model, a CNN architecture for multiresolution volumetric segmentation from 4D data. Input data were 4D-CTA subvolumes consisting of 19 timepoints and an input size of $n \times n \times n$ voxels, with $n \in \{92, 100, 108, 116, \dots\}$. The network produced volumetric class probability maps for the four segmentation classes with the same size as the input minus the size of the receptive field of $88 \times 88 \times 88$ voxels. BN is an abbreviation for batch normalization.

2.2 Methods

2.2.1 Model architecture

CNNs can be represented as directed graphs, where a node (hereinafter referred to as layer) indicates an operation on volumetric feature maps, incoming edges indicate what feature maps are fed to a layer, and outgoing edges represent the feature maps produced by a layer. Figure 2.1 shows such a representation of our model. It consists of 15 convolution layers (green), 3 max-pool layers (yellow), and 3 upscale layers (blue). All solid arrows form the path through the network that visits all the layers exactly once, whereas the dotted arrows skip several layers within the network (shortcuts). In addition to the shortcuts from the original U-Net[51], shortcuts were added over every two consecutive 3^3 convolutions as these were found to speed up convergence and increase overall performance in combination with the other shortcuts[52].

The model uses concatenation or elementwise summation to merge two sets of feature maps at a layer into a single set of feature maps. The concatenation layer joined the two sets, resulting in a larger set of feature maps. To perform elementwise summation, both sets are required to have the same number of feature maps. If this was not the case, all feature maps from the first set A (at the start of the curved arrows in Figure 2.1) were (repeatedly) iterated and concatenated to a new set C until the number of feature maps between set A and the second set B were the

same. Next, the new set C was used instead of the original first set A to perform elementwise summation. For example, let $A = (a, b, c)$ (3 feature maps) and $B = (f_1, f_2, f_3, \ldots, f_{64})$ (64 feature maps). Now a new set $C = (a, b, c, a, b, c, \ldots)$ is created from set A by iterating its elements until it has the same 64 feature maps as set B. Finally, the feature map sets B and C are merged by summation per feature map at the summation layer $D = (f_1 + a, f_2 + b, f_3 + c, f_4 + a, f_5 + b, \ldots, f_{63} + c, f_{64} + a)$. Note that swapping the contents of sets A and B, yields $D = (f_1 + a, f_2 + b, f_3 + c)$. The feature maps in both sets were cropped around the center to the smallest input feature map size for both merge layers to resolve size mismatches.

The network was inspired by the 3D U-Net architecture. Feature extraction at each voxel resolution was achieved by two subsequent 3^3 convolution layers with batch normalization[53]. Each of these convolutions was followed by a leaky variant of a rectified linear activation unit (very leaky ReLU) as defined in[54]:

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ x/3, & \text{otherwise} \end{cases} \tag{2.1}$$

The very leaky ReLU was preferred over the normal ReLU since it emits similar behavior, but prevents 'dying ReLU'. This problem refers to a unit that only produces zeros for any given input and which is unlikely to break out of that state during training, which makes these units no longer useful. Downscaling the input by a factor of 2 was achieved by a 2^3 max-pool layer. For this architecture, there are four voxel resolutions at which features were extracted: the original resolution of 0.5 mm^3/voxel, 1.0 mm^3/voxel, 2.0 mm^3/voxel, and 4.0 mm^3/voxel. To synthesize the output class probability maps from the lower resolution feature maps, the lower resolution feature maps were first upsampled at each upsampling layer by a factor 2 using nearest neighbor interpolation and were then concatenated with the feature maps acquired earlier at a similar resolution (depicted by the horizontal striped arrows in Figure 2.1). This upscaling operation was preferred to the deconvolution operations in Çiçek et al.[40], since the latter is thought to introduce artificial checkerboard patterns in the output[55]. This data integration process was repeated from the lowest to the highest resolution until feature maps at the original voxel resolution were retrieved. Finally, the output at the last layer was passed through a soft-max activation function.

The network architecture was fixed for training and evaluation and therefore introduced a fixed relation between network input size and network output size. For instance, at each 3^3 convolution layer, the size of the input feature maps is reduced by 2 voxels, whereas a 2^3 max-pooling layer halves the number of voxels and a 2^3 upsampling layer doubles the spatial voxel size for the output feature maps. As the feature maps were passed from layer to layer through the network, it finally produced

an output size that had 88 voxels less than the input size in every spatial dimension. In this particular case, the size difference equals the size of the receptive field of the network, where the size of the receptive field of the network is the spatial extent of the input voxels (subvolume) which contribute to the activation of a single output unit, i.e., to the output class probability for an individual voxel.

The network input were batches consisting of 4D-CTA subvolumes. Each subvolume could be varied in size (number of voxels per spatial dimension) and could be varied in batch size (number of subvolumes per batch), but should always have a fixed number of timepoints. Selecting the subvolume size and the batch size have practical implications on the required GPU RAM and on training performance. Valid input size values are $n \in \{92, 100, 108, 116, \dots\}$, since n must be bigger than the size of the receptive field of the network ($n > 88$) and the input size should produce even-sized feature maps before each pooling layer to preserve voxel correspondence at each resolution. For the experiments in this work, we fixed the number of timepoints to 19, since it matched the number of timepoints for each 4D-CTA acquisition collected for this study (see section 2.3). For network training, we put the subvolume size to $n = 124$ voxels for each spatial dimension and employed a batch size of 2. This gave an output class probability map per segmentation class with 36 voxels $(124 - 88)$ for each spatial dimension.

The full-size final prediction segmentations were obtained following a similar strategy as described by Çiçek et al.[40], Ronneberger et al.[51], by repeatedly shifting and applying a CNN on the input data until all input voxels had their corresponding predictions. First, the input data was zero-padded with a border half the size of the receptive field of 44 voxels for all spatial dimensions. Next, the model was repeatedly applied until all voxels within the input data had corresponding brain tissue predictions.

2.2.2 Model training

In deep learning, training of the architecture is at least as important as the design of the architecture. In this work, a training strategy was used similar to the work of Çiçek et al.[40], consisting of a categorical cross-entropy objective function adapted for sparse data; this was minimized using default stochastic gradient descent optimizer with Nestorov momentum[56]. Training was done on sparse annotations, that is, annotations in 2D cross sections of 3D volumes derived from 4D data (See section 2.3). In this section, we describe the objective function and parameter regularization, data sampling and augmentation, parameter initialization, optimizer, and other technical details. The reported hyperparameters in this section were experimentally

selected.

Objective and regularization

The training objective is to find the set of weight parameters Θ for our model that minimizes the loss function $L(\Theta \mid t, w)$, given the reference standard t and voxel weights w. The loss function was constructed from the weighted categorical cross entropy $WCCE(\cdot)$, and L_1-norm $L_1(\cdot)$ and L_2-norm $L_2(\cdot)$ weight regularization terms, as follows:

$$\begin{aligned} L(\Theta \mid t, w) =& \lambda_0 WCCE(\Theta \mid t, w) + \\ & \lambda_1 L_1(\Theta) + \lambda_2 L_2(\Theta) \end{aligned} \tag{2.2}$$

where $\lambda_0 = 1, \lambda_1 = 1e^{-6}$, and $\lambda_2 = 1e^{-5}$. The $WCCE(\cdot)$ is the weighted categorical cross-entropy loss function, which calculates the weighted mean over the categorical cross entropy $CCE_i(\cdot)$ per voxel i with weights w_i. The $CCE_i(\cdot)$ defines the error between output $p_{i,j}(\Theta)$ of the soft-max activation function at the last layer of our model given the weight parameters Θ and the reference standard $t_{i,j}$ for each voxel i and segmentation class j:

$$\begin{aligned} WCCE(\Theta \mid t, w) &= \frac{\sum_i w_i CCE_i(\Theta \mid t)}{\sum_i w_i} \\ CCE_i(\Theta \mid t) &= -\sum_j t_{i,j} log(p_{i,j}(\Theta)) \end{aligned} \tag{2.3}$$

The weights w were set to an annotation mask by setting the weights w_i to 1 if annotations were present for voxel i and to 0 otherwise, thereby only learning from labeled voxels.

Dropout was applied during training before the 3^3 convolutions by setting 50% randomly selected voxels to zero at the coarsest image resolution (Figure 2.1) for each processed batch.

Sampling

All the annotated voxels within the cranial cavity formed the sampling candidates. The cranial cavity is defined as the space containing all soft tissues and CSF, including the meninges, cerebrum, ventricles, cerebellum, and brain stem, and was segmented using the method of Patel et al.[18]. Each subvolume selected during training was centered on a single sampling candidate in world coordinates. All subvolume voxels that were sampled outside of the input data were set to zero value.

Each CNN model was shown 60k subvolumes, which were processed in batches of 2 subvolumes during training. For every 400 subvolumes, an equal number of candidates were sampled uniformly per tissue type from the set of sampling candidates.

Augmentations

Five types of augmentations were used during training to artificially enlarge the sparsely annotated dataset. The use of augmentations has been shown to prevent overfitting, improve generalization, and introduce invariance to the augmentations used[8,40].

For each subvolume in the training data, one of the five following augmentations was assigned with equal probability: identity-, mirroring-, rotation-, uniform scaling-, or elastic deformation. Only one augmentation was computed per subvolume to keep the computation time low. The identity transformation reproduces the original signal. Mirroring flips the input along the sagittal axis only. Rotation is expressed as a 3D Euler rotation in degrees around the center of the subvolume where the $x, y,$ and z rotations are individually sampled from the continuous uniform distribution $\mathcal{U}(-8, 8)$. Uniform scaling is defined as an affine transform that rescales the input uniformly by a scalar over all axes, which is sampled from the continuous uniform distribution $\mathcal{U}(1.01, 1.25)$. Scaling down was omitted from the scaling augmentation since it could potentially remove small vascular structures in the input. The elastic deformation applies a 3D linear interpolation of the input subvolume where each individual corner point of the bounding box of the subvolume was given a different randomized offset in voxels for the $x, y,$ and z coordinates drawn from the normal distribution $\mathcal{N}(0, 6)$, resulting in warped subvolumes.

A selected transformation was calculated once and then applied to the input subvolume, annotation labels, and annotation mask, with interpolation orders 1, 0, and 0, respectively.

Weight initialization

At the start of training, all weights in the model were initialized using a He initialization scheme[57], which was adjusted for the very leaky ReLU activation function (equation 2.1). That is, at each layer, the weights were sampled from the following normal distribution:

$$\mathcal{N}\left(0, \sqrt{9/(5fan_{in})}\right) \tag{2.4}$$

where fan_{in} is defined as the number of feature maps being input to the layer multiplied by the size of the convolution kernel. To keep the initialization constant across

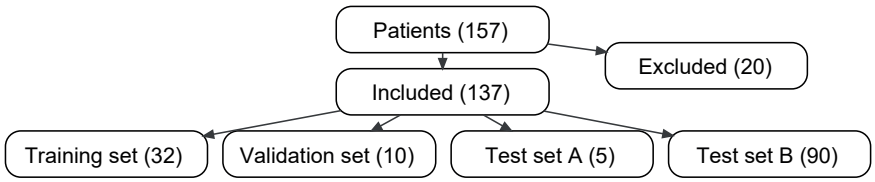


Figure 2.2: Data flow diagram for the 4D-CTA data distribution over the training set, the validation set, test set A, and test set B.

different experiments, the random generators were seeded with the same constant.

Optimizer and implementation

A stochastic gradient descent optimizer was used starting with a learning rate of 0.1, which was decreased by a factor of 10 after having processed every 20k subvolumes. Momentum was used and was kept constant at 0.9.

The model was implemented in Theano/Lasagne[58,59] and training was performed on an NVidia Titan X graphics card with 12 GB of video memory.

2.3 Data

2.3.1 Patient inclusion and image acquisition

This retrospective study was approved by the institutional ethics committee and informed consent was waived. In total, 157 patients (age 63 ± 14 years, 61% male) with a suspicion of stroke in 2015 or 2016 at the Radboud University Medical Center, Nijmegen, the Netherlands, were included. 4D-CTA were acquired using a 320-row CT scanner (Toshiba Aquilion ONE, Japan) consisting of 19 volumetric scans with different exposures per timepoint. Patients received 80 mL of contrast agent (Iomeron) injected in the cephalic vein at the start of the first acquisition. Image reconstruction was done using an FC41 smooth convolution kernel, resulting in $512 \times 512 \times 320$ voxels with a voxel size of $0.47 \times 0.47 \times 0.5$ mm. One full 4D CT acquisition took in total less than a minute to complete using a strict protocol with fixed time intervals between each of the 19 volumetric scans. No preprocessing or motion correction were performed during the acquisition.

Twenty patients were excluded because of the presence of large pathology (bleedings, infarcts, and excessive liquor) or because of imaging artifacts (e.g., clips, drains, patient motion, or beam hardening). Test set B was formed from ninety patient cases.

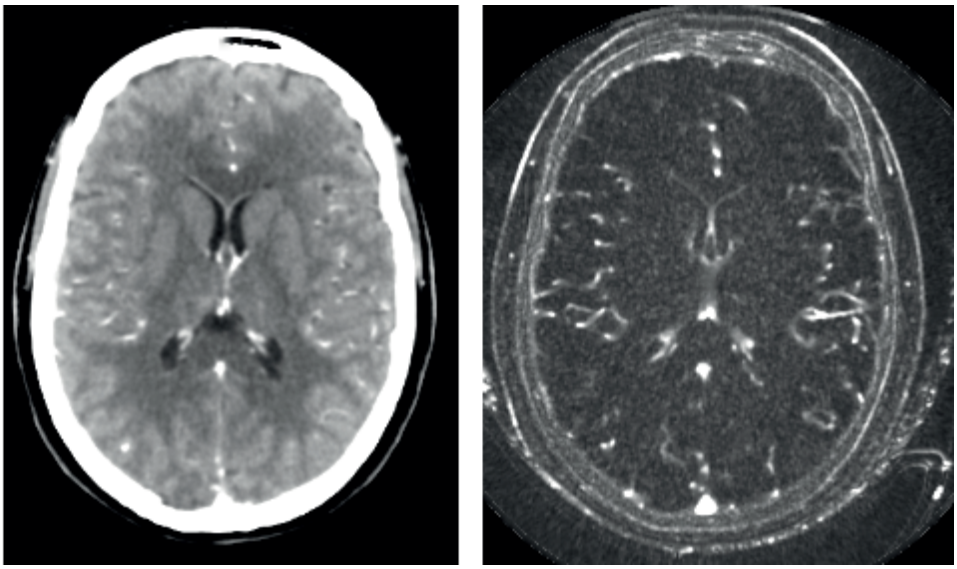


Figure 2.3: Example axial cross section for the derived images of a single 4D-CTA image used for annotation. Left: the temporal average for WM, GM, and CSF segmentation. Right: the temporal variance for vessel segmentation.

The remaining 47 patients were randomly split into a training set of 32, a validation set of 10, and test set A of 5 patients. Test set A was also used to assess the observer variability. Figure 2.2 summarizes the data selection.

2.3.2 Preprocessing

Timepoints $t > 0$ were rigidly registered to the first timepoint ($t = 0$), to correct for potential head movement during acquisition. The registration was performed with Elastix[60] using the steps and parameter settings as described by Manniesing et al.[30].

Cerebral soft tissue has a limited intensity range in CT, approximately 20 HU to 65 HU[61]. Intensity values outside this range, for example, bone that starts from 700 HU, may complicate CNN training and limit the optimal achievable performance. Therefore, the registered 4D-CTA was first clipped within the range $[-50, 400]$ then linearly scaled to $[0, 1]$. A broader clipping range was used rather than the defined soft tissue HU ranges to preserve more spatial contextual information.

2.3.3 Reference standard

The reference standard was obtained by manually annotating the WM, GM, CSF, and vessels in a single 2D cross-section per patient imaging data. Annotations were carried out by two medical students, who were trained and supervised by an experienced neuroradiologist with more than 10 years of experience. A 3D annotation tool, called VCAST (volumetric cluster annotation and segmentation tool[62]) was developed in-house specifically for this task. VCAST provides normal annotation capabilities (like brushes for annotating voxels in a cross-section) and, in addition, provides supervoxel grids of various sizes to add or remove annotations. Other capabilities of the tool include instant navigation to the cross sections requiring annotations and preset keys for the window levels (center/width was 30/80 HU for CSF, 50/50 HU for WM/GM, and 60/60 for vessels).

4D-CTA data is hard to interpret by human readers, which results in long annotation times and an increased likelihood of error. To facilitate the human readers, two images were derived for each 4D-CTA, by merging the temporal information. The weighted temporal average (WTA)[30] for annotating the WM, GM, and CSF because it has the highest signal-to-noise ratio and best soft tissue contrast and the weighted temporal variance (WTV)[63] for annotating the vessels because of its sensitivity to contrast variations. However, even in the WTV image, manually annotating vessel structures is complex and time-consuming because of their varying shapes and sizes and the partial volume effects. Therefore, vessels were first pre-segmented by an automated segmentation algorithm based on local histogram features and a random forest classifier[63]. This segmentation was then presented within VCAST for further manual refinements. See Figure 2.3 for an example of the derived images.

The areas selected for annotation are indicated in blue in Figure 2.4. The cerebellum was insufficiently detailed for an experienced reader to reliably derive WM and GM annotations from — mainly because of the limitations of CT imaging — and was therefore excluded from all cross sections. The falx cerebri and the tentorium cerebelli were left out because these structures do not contain any of the four tissue types used in this study.

The method of Patel et al.[18] was used to segment all intracranial soft tissue, which was then manually adjusted to reflect masks similar to Figure 2.4. For each cross-section, the orthogonal plane was randomly selected, after which the cross-section to be annotated was extracted from the selected plane. All cross sections consisting of less than 10% of the mask voxels were excluded from selection. Six patients had 2D cross-sections for all three orthogonal planes, each plane was selected using previously described method.

Figure 2.4: Three cross sections (axial, coronal, sagittal) of an exemplar 4D-CTA case. Blue areas were selected for annotation by the observers, other areas were not annotated.

During annotations, four small densely connected voxel subareas were found within the annotated cross sections for which the soft tissue labels could not reliably be determined by the observers; these areas were ignored during training (three areas) and evaluation (one area). The areas had an average size of 76.4 mm^3, were less than 0.1% of all annotated voxels, and the effect on the evaluation measures was assessed to be insignificant. After the observers annotated all patients once, two qualitative inspections were performed by the radiologist to assess overall annotation quality. Errors detected during these inspections were subsequently corrected.

In total, over 410 hours were spent by the observers in creating the reference standard.

2.4 Experiments

2.4.1 Observer variability

Observer variability was estimated on five 4D-CTA data from test set A, which were annotated in two subsequent series by both observers. When observers were unsure about their annotations, they were asked to leave those voxels out. Only voxels annotated twice by both observers were used for calculating the estimation. Intraobserver variability was reported for both observers and interobserver variability was reported for the first series of annotations. The Dice Similarity Coefficient (DSC)[64], contour mean distance (CMD), absolute volume difference (AVD), and mean volume difference (MVD) were used as measures of evaluation.

The CMD between two non-zero pixel sets A and B is defined as the mean dis-

tance between boundaries of non-zero pixel regions:

$$CMD(A,B) = max(h(A,B), h(B,A))$$
$$h(A,B) = \text{mean}_{a \in A} \min_{b \in B} ||a - b||$$

The MVD between two non-zero-pixel sets A and B computes the volume difference in mm^3 and is defined as:

$$MVD(A,B) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} |A_i - B_j|$$

The AVD, which computes the relative volume difference between A and B in $\%$, is subsequently defined as:

$$AVD(A,B) = \frac{MVD(A,B)}{\frac{1}{n}\sum_{i=1}^{n} A_i}$$

2.4.2 Model evaluation

Our model was compared with 3D U-Net[40], which is a state-of-the-art CNN model for volumetric image segmentation, on DSC for the segmentations. The models are similar except that our model has additional shortcuts over every pair of two 3^3 convolutions, uses very leaky ReLU instead of ReLU as an activation function throughout the architecture, and uses nearest neighbor upsampling instead of deconvolution. Concerning the training parameters, our model uses modified He initialization instead of Xavier initialization[65], uses a batch size of 2 instead of 1 and has additional L_1 and L_2 regularization terms on the weights.

For a fair comparison, the 3D U-Net was trained and evaluated in the same manner as described in section 2.2.2, but used the architecture and weight initialization scheme from the original work. Additionally, the two models shared most of the hyperparameters, which were experimentally tuned on the validation set. We have kept the batch size (2) and subvolume size (124^3 voxels) the same. Also, upsampling layers instead of deconvolution layers were used since we wanted to avoid checkerboard artifacts[55]. Furthermore, we used the same optimizer, learning rate scheme, momentum term, L1 and L2-norm weighting, and augmentations from section 2.2.2. Essentially, the only differences between the models were: the weight initialization, the use of additional shortcuts over every pair of two 3^3 convolutions, and the activation function. All other hyperparameters were kept the same.

DSC, CMD, AVD, and MVD were used as evaluation measures in all experiments. Each model was trained on 60k subvolumes randomly drawn from the annotated voxels of the training set of 32 registered and normalized cases (section 2.3.2). After

every 400 subvolumes processed the DSC for each model was calculated on the separate validation set of 10 cases. After completing all iterations, the model with the highest average DSC for all classes on the validation set was evaluated one last time on test set A of five cases to assess the final performance. The averages of all the test cases were reported and specified per tissue type and observer.

2.4.3 State-of-the-art comparison

Our best-performing model from section 2.4.2 was compared to Manniesing et al.[30] which is the current state-of-the-art for WM/GM segmentation in 4D-CTA. The latter method is based on feature extraction and support vector machine (SVM) classification. It was evaluated on a different dataset with 22 different patients than the 32 patients in section 2.3, but the data was obtained with the same scanner. The dataset had more annotated cross-sections 87 than our training dataset 40. The method was cross-validated on selected axial cross-sectional slices in 22 patients.

For a fair comparison, only the voxels within the WM and GM classes defined by their reference standard were used for evaluation, since the competing method provided coarse unevaluated segmentation classes for CSF and vessels. We compared the output segmentations of our method with that of Manniesing et al.[30] using DSC, CMD, AVD, and computation time. The best-performing model (full model, trained on 4D data from scratch) was selected and applied to the entire dataset from Manniesing et al.[30] without preprocessing or fine-tuning. Similar to Manniesing et al.[30], the statistics were first calculated per slice and then averaged over all slices.

2.4.4 Extended evaluation

Our method was applied to all cases from test set B and the resulting segmentations were qualitatively inspected. For ten of these cases, a single cross-section was annotated by a single observer using the same selection and annotation procedures as in section 2.3.3. For this annotated subset, segmentations from our method and 3D U-Net were quantitatively scored on DSC, CMD, AVD, and MVD.

2.4.5 Ablation experiments

Ablation experiments were performed to assess the contribution of the He initialization scheme versus Xavier initialization, the addition of the shortcut connections, and the replacement of the ReLU by the very leaky ReLU activation function. We used our best-performing model architecture and training scheme as a basis and trained

Table 2.1: The observer variability across five cases. Measures used are the Dice coefficient (DSC), contour mean distance (CMD) in mm, and absolute volume difference (AVD), and mean volume difference (MVD) in mm^3.

		CSF	WM	GM	vessel	mean
DSC	**intraobs. 1**	0.94 ± 0.01	0.94 ± 0.01	0.95 ± 0.01	0.95 ± 0.01	0.94 ± 0.01
	intraobs. 2	0.85 ± 0.07	0.87 ± 0.02	0.91 ± 0.02	0.90 ± 0.03	0.88 ± 0.05
	interobs.	0.86 ± 0.04	0.86 ± 0.02	0.89 ± 0.02	0.89 ± 0.03	0.88 ± 0.03
CMD	**intraobs. 1**	0.13 ± 0.10	0.10 ± 0.02	0.13 ± 0.06	0.08 ± 0.02	0.11 ± 0.07
(mm)	**intraobs. 2**	1.16 ± 1.26	0.24 ± 0.08	0.37 ± 0.13	0.42 ± 0.17	0.55 ± 0.73
	interobs.	1.46 ± 1.61	0.35 ± 0.13	0.48 ± 0.09	0.59 ± 0.20	0.72 ± 0.93
AVD	**intraobs. 1**	2.48 ± 1.62	2.56 ± 2.55	2.35 ± 2.95	4.52 ± 2.64	2.98 ± 2.65
(%)	**intraobs. 2**	6.89 ± 6.87	6.98 ± 4.28	3.61 ± 2.88	3.78 ± 2.65	5.31 ± 4.78
	interobs.	17.28 ± 9.59	6.81 ± 2.50	6.81 ± 3.61	4.52 ± 4.94	8.86 ± 7.65
MVD	**intraobs. 1**	17 ± 9	61 ± 76	48 ± 71	10 ± 7	34 ± 57
(mm^3)	**intraobs. 2**	68 ± 63	120 ± 64	61 ± 48	7 ± 5	64 ± 65
	interobs.	152 ± 97	129 ± 56	116 ± 66	11 ± 12	102 ± 85

three new models. For the first model, we replaced the modified He initialization scheme with Xavier initialization. For the second model, we left out the additional short shortcut connections, and for the third model, we replaced the very leaky ReLU functions with normal ReLU functions. All models were reinitialized at the beginning of training and were trained as described in section 2.2.2. The best models were selected by taking the highest average DSC performance on the validation set. The best models were evaluated on the ten annotated cases from the previous experiment.

2.5 Results

2.5.1 Observer variability

The observer results are summarized in Table 2.1. The average DSC intra- and interobserver agreements were equal or greater than 0.85 for all tissue types for both observers, with most classes having over 0.90 overlap. Interobserver agreement had average DSC scores equal to or greater than 0.86 and overall were slightly lower than the intraobserver agreement. Paired t-test showed statistically significant differences ($p < 0.05$) between the two observers for all tissue types.

2.5.2 Model evaluation

The evaluation results are summarized in Table 2.2. In general, high degrees of overlap with our model and the reference standard were found for all classes, with average DSC in the range of $[0.85, 0.88]$ for observer 1 and $[0.82, 0.84]$ for observer 2. Paired t-tests over all experiments and classes showed significant differences between observers ($p < 0.05$). Paired t-tests showed that the segmentation results from our model and that of 3D U-Net differed significantly ($p < 0.05$). The training time for each of the models was approximately 4 days.

Figure 2.5 shows the results on test set A of five patients obtained from the best performing model.

2.5.3 State-of-the-art comparison

The comparison results are summarized in Table 2.3. Paired t-test showed significant differences for all three evaluation measures for GM and computation time ($p < 0.05$) and a significant difference for WM on CMD ($p < 0.05$). In general, our model outperforms the pattern recognition SVM method by Manniesing et al.[30] on DSC, AVD, CMD and computation time.

2.5.4 Extended evaluation

The segmentations from our method show good differentiation of the WM, GM, CSF, and Vessels, with a slight overestimation of the GM. The method makes more mistakes around imaging artifacts, like streaking and metal artifacts, but overall these errors appear minor. The quantitative results on the extended evaluation set are listed in Table 2.4. Paired t-tests showed significant differences between our model and 3D U-Net, for all tissue types and all metrics ($p < 0.05$). Overall, our model outperforms 3D U-Net on all metrics.

2.5.5 Ablation experiments

The ablation results are listed in Table 2.5. Paired t-tests showed significant differences, for all tissue types and all metrics, between our model and our model without additional shortcuts over 3^3 convolution pairs and between our model and our model with ReLU instead of very leaky ReLU activation functions ($p < 0.05$). However, the tests showed no significant differences between our model and our model with Xavier initialization instead of He initialization ($p > 0.05$).

Table 2.2: Quantitative segmentation results on the observer reference standards for our model and 3D U-Net. The Dice coefficient (DSC), contour mean distance (CMD) in mm, absolute volume difference (AVD) in %, and mean volume difference (MVD) in mm^3 were used for which the mean and standard deviation were calculated for all five cases in test set A per tissue type and per observer (obs 1 and obs 2). For comparison, we have added the interobserver variability. Paired t-tests showed significant differences between observers $p < 0.05$ and between models $p < 0.05$, see section 2.5.2 for details. Bold values indicate the best performance between models per metric, per class, and per observer.

		interobs.	**our model**		**3D U-Net**	
			vs obs 1	**vs obs 2**	**vs obs 1**	**vs obs 2**
DSC	**CSF**	0.86 ± 0.04	$\mathbf{0.85 \pm 0.05}$	$\mathbf{0.81 \pm 0.06}$	0.76 ± 0.10	0.75 ± 0.09
	WM	0.86 ± 0.02	$\mathbf{0.88 \pm 0.04}$	$\mathbf{0.86 \pm 0.03}$	0.88 ± 0.03	0.86 ± 0.03
	GM	0.89 ± 0.02	$\mathbf{0.88 \pm 0.02}$	$\mathbf{0.84 \pm 0.02}$	0.85 ± 0.03	0.81 ± 0.03
	vessel	0.89 ± 0.03	$\mathbf{0.86 \pm 0.03}$	$\mathbf{0.83 \pm 0.03}$	0.65 ± 0.11	0.64 ± 0.10
	mean	0.88 ± 0.03	$\mathbf{0.87 \pm 0.04}$	$\mathbf{0.84 \pm 0.04}$	0.78 ± 0.12	0.77 ± 0.11
CMD	**CSF**	1.46 ± 1.61	$\mathbf{0.82 \pm 0.80}$	$\mathbf{0.65 \pm 0.45}$	1.68 ± 1.45	1.22 ± 0.93
(mm)	**WM**	0.35 ± 0.13	$\mathbf{0.49 \pm 0.15}$	$\mathbf{0.70 \pm 0.14}$	0.65 ± 0.21	0.76 ± 0.13
	GM	0.48 ± 0.09	$\mathbf{0.38 \pm 0.24}$	$\mathbf{0.38 \pm 0.08}$	0.43 ± 0.20	0.40 ± 0.15
	vessel	0.59 ± 0.20	$\mathbf{0.39 \pm 0.16}$	$\mathbf{0.58 \pm 0.23}$	4.17 ± 1.69	4.20 ± 1.65
	mean	0.72 ± 0.93	$\mathbf{0.52 \pm 0.47}$	$\mathbf{0.58 \pm 0.29}$	1.73 ± 1.86	1.64 ± 1.78
AVD	**CSF**	17.28 ± 9.59	$\mathbf{12.95 \pm 12.24}$	$\mathbf{12.15 \pm 7.49}$	48.90 ± 40.78	35.76 ± 29.74
(%)	**WM**	6.46 ± 2.15	$\mathbf{10.62 \pm 6.97}$	13.22 ± 4.85	13.03 ± 4.74	$\mathbf{12.36 \pm 6.64}$
	GM	6.48 ± 3.46	9.28 ± 7.27	12.61 ± 6.58	$\mathbf{3.58 \pm 3.69}$	$\mathbf{5.41 \pm 3.63}$
	vessel	4.35 ± 4.76	$\mathbf{14.26 \pm 9.93}$	$\mathbf{14.50 \pm 11.15}$	27.41 ± 14.63	30.10 ± 14.34
	mean	8.86 ± 7.65	$\mathbf{11.78 \pm 9.55}$	$\mathbf{13.12 \pm 7.91}$	23.23 ± 27.75	20.91 ± 21.01
MVD	**CSF**	152 ± 97	$\mathbf{74 \pm 41}$	$\mathbf{109 \pm 82}$	267 ± 66	226 ± 79
(mm^3)	**WM**	129 ± 56	$\mathbf{165 \pm 110}$	208 ± 61	209 ± 72	$\mathbf{188 \pm 85}$
	GM	116 ± 66	192 ± 149	267 ± 161	$\mathbf{61 \pm 55}$	$\mathbf{111 \pm 73}$
	vessel	11 ± 12	$\mathbf{27 \pm 19}$	$\mathbf{28 \pm 18}$	50 ± 23	59 ± 26
	mean	102 ± 85	$\mathbf{114 \pm 116}$	153 ± 133	147 ± 110	$\mathbf{146 \pm 95}$

2.6 Discussion

We have presented a fully convolutional multiclass deep learning architecture for 3D segmentation which can learn end-to-end from sparsely annotated 4D data. The method gives high-quality segmentations of WM, GM, CSF, and vessels in 4D-CTA, approximating the interobserver agreement and outperforms the current state-of-the-art.

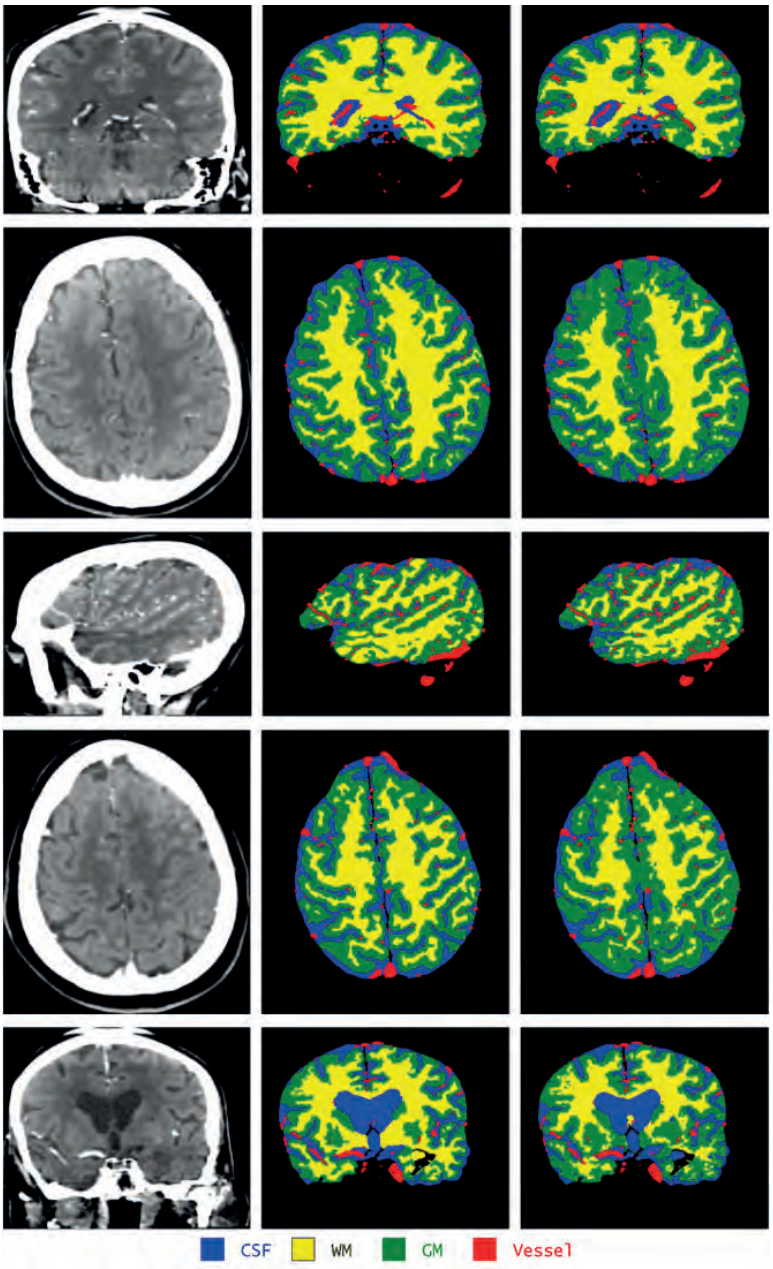


Figure 2.5: Qualitative results produced by our model on test set A. From left to right: temporal average, reference standard (observer 2), model prediction; each row represents an annotated cross-section from the five test cases. The cerebellar area in the top row was unlabeled.

Table 2.3: Comparison between our model and Manniesing et al.[30] on Dice coefficient (DSC), contour mean distance (CMD) in mm, and absolute volume difference (AVD) in %. The first three rows show segmentation scores for white matter (WM) and gray matter (GM). The final row shows the average computation time of the segmentation per model. * indicates a $p < 0.05$. Bold values indicate the best performance between models per metric and per class.

	Ours		**Manniesing et al.[30]**	
	WM	**GM**	**WM**	**GM**
DSC	$\mathbf{0.82 \pm 0.07}$	$\mathbf{0.81 \pm 0.04}^*$	0.81 ± 0.04	0.79 ± 0.05
CMD (mm)	$\mathbf{0.94 \pm 0.40}^*$	$\mathbf{0.57 \pm 0.30}^*$	1.35 ± 0.26	0.74 ± 0.19
AVD (%)	$\mathbf{13.54 \pm 11.83}$	$\mathbf{9.80 \pm 6.62}^*$	15.83 ± 10.85	16.48 ± 11.16
Time	$\pm$**5 mins***		$\pm$60 mins	

The experimental results (Table 2.2 and Table 2.4) highlight that our model significantly outperforms 3D U-Net with respect to the DSC and CMD. This is likely to be the combined contribution of the additional shortcuts, and the very leaky ReLU activation function (Table 2.5). The additional shortcuts are thought to simplify learning by allowing information to directly skip the 2^3 convolutions pairs throughout the network. The very leaky ReLU activation function is thought to work better since it avoids 'dead' ReLU, which is a state of a normal ReLU that always outputs zero and is unlikely to break out of this state. Finally, He initialization was thought to work better than Xavier initialization, since it has been optimized for the ReLU function, which we use throughout the network. However, there was no significant improvement found from the ablation experiments (Table 2.5).

Our model slightly, but significantly, outperforms the current state-of-the-art method by Manniesing et al.[30] on WM and GM segmentation with respect to the DSC, AVD, and CMD (see Table 2.3), without any training or optimization on the dataset used to train the competing method and with our model being trained on less annotated training slices. Despite these disadvantages our model significantly outperformed the competing state-of-the-art method. If such measures were taken the model is expected to perform even better. Furthermore, at prediction time, our model can be run on a GPU within 5 minutes whereas the competing method, which can not be easily GPU optimized, takes hours to compute on multiple CPUs. Additionally, our model provides CSF and vessel segmentations that were learned directly from 4D-CTA data as opposed to unevaluated segmentation methods based on simple heuristics.

The quantitative results (see Table 2.2) approximate the interobserver overlap for the model compared to observer 1. However, while still having good overlap, the

Table 2.4: Quantitative segmentation results on the extended reference standard for our model and 3D U-Net. The Dice coefficient (DSC), contour mean distance (CMD) in mm, absolute volume difference (AVD) in %, and mean volume difference (MVD) in mm^3 were used for which the mean and standard deviation were calculated for the ten annotated cases in test set B per tissue type. Paired t-tests showed significant differences between models $p < 0.05$. Bold values indicate the best performance between models per metric and per class.

		our model	3D U-Net
DSC	**CSF**	$\mathbf{0.82 \pm 0.07}$	0.67 ± 0.14
	WM	$\mathbf{0.85 \pm 0.02}$	0.78 ± 0.03
	GM	$\mathbf{0.82 \pm 0.02}$	0.75 ± 0.03
	vessel	$\mathbf{0.81 \pm 0.05}$	0.51 ± 0.17
	mean	$\mathbf{0.82 \pm 0.04}$	0.68 ± 0.15
CMD	**CSF**	$\mathbf{0.61 \pm 0.27}$	1.73 ± 1.01
(mm)	**WM**	$\mathbf{0.92 \pm 0.31}$	1.88 ± 0.54
	GM	$\mathbf{0.66 \pm 0.19}$	0.88 ± 0.29
	vessel	$\mathbf{0.72 \pm 0.39}$	6.82 ± 2.47
	mean	$\mathbf{0.73 \pm 0.30}$	2.83 ± 2.71
AVD	**CSF**	$\mathbf{10.57 \pm 4.71}$	62.68 ± 48.45
(%)	**WM**	$\mathbf{13.23 \pm 7.60}$	23.82 ± 12.62
	GM	19.29 ± 11.81	$\mathbf{14.56 \pm 11.27}$
	vessel	$\mathbf{15.00 \pm 9.41}$	44.18 ± 21.60
	mean	$\mathbf{14.52 \pm 8.69}$	36.31 ± 31.90
MVD	**CSF**	$\mathbf{108 \pm 61}$	509 ± 418
(mm^3)	**WM**	$\mathbf{237 \pm 118}$	487 ± 358
	GM	307 ± 198	$\mathbf{210 \pm 135}$
	vessel	$\mathbf{18 \pm 12}$	51 ± 23
	mean	$\mathbf{168 \pm 159}$	314 ± 327

results are significantly inferior for the model compared to observer 2. This difference might be a result of the fact that two out of three training cases were annotated by observer 1. In other words, there was an observer imbalance of the training set. Another explanation is that observer 1 had significantly lower intraobserver variability, which may have resulted in easier cases for the model to generalize to.

The scores from the extended evaluation on the test set B (see Table 2.4) are overall in line with the findings on test set A (see Table 2.2).3D U-Net significantly performed worse than our model. However, the results on test set B are worse on average than those on test set A. We suspect that to be the case because test set B had more difficult cases than test set A. Because of this, the previously obtained

Table 2.5: Ablation experiment segmentation results on Dice coefficient (DSC), contour mean distance (CMD) in mm, absolute volume difference (AVD) in %, and mean volume difference (MVD) in mm^3. Reported values are averages over all tissue classes. From left to right: our best performing model (Ours), our model with Xavier instead of He initialization (Ours-Xavier), our model with additional shortcuts over 3^3 convolution pairs removed (Ours-no skip), and our model with ReLU instead of very leaky ReLU activation functions (Ours-ReLU). * indicates a significant difference between the average metric score for our model and the average metric score of the ablated model ($p < 0.05$). Bold values indicate the best performance between models per metric.

	Ours	Ours-Xavier	Ours-no skip	Ours-ReLU
DSC	$\mathbf{0.82 \pm 0.04}$	0.80 ± 0.11	$0.81 \pm 0.06^*$	$0.70 \pm 0.12^*$
CMD	$\mathbf{0.73 \pm 0.30}$	1.27 ± 1.29	$0.93 \pm 0.51^*$	$1.55 \pm 1.22^*$
AVD	$\mathbf{14.52 \pm 8.69}$	17.23 ± 18.29	$21.94 \pm 11.59^*$	$43.77 \pm 52.79^*$
MVD	168 ± 159	$\mathbf{99 \pm 74}$	$268 \pm 253^*$	$469 \pm 564^*$

interobserver overlap on test set A cannot be fairly compared with the new scores, since the previous indications are expected to be overly optimistic regarding the more difficult cases. Furthermore, we cannot compute new indications based on a single observer.

We emphasize that annotating brain tissues of 4D-CTA is a very difficult task for humans. Even though the observers had access to 3D data while annotating, in practice they tended to focus mainly on a single 2D cross-section, which may introduce an annotation bias. Visualization of test set A predictions (see Figure 2.5) generally resulted in a good approximation of the reference standard. In the axial cross sections, some GM over-segmentation occurred with respect to the reference standard, but the model predictions seem to better match the underlying anatomy presented in the temporal average image.

The dataset used for the evaluation consisted exclusively of normal-appearing brain tissues without pathology or foreign objects, which are seen in everyday clinical practice. The data was collected as such to focus on testing the feasibility of segmentation of WM/GM/CSF and vessels in 4D-CTA using deep learning, which is traditionally the domain of MR imaging. This implies that the method likely must be trained on cases with pathology or foreign objects and at least be evaluated on such cases before it can be used in practice. However, we argue that our method provides a valuable first step towards this goal in the next paragraph.

In principle, the architecture is not limited to normal tissues. It can easily be extended to include tissue classes for pathology or foreign objects, such as core,

penumbra, bleedings, clips, drains, calcifications, and bone if sufficiently annotated data for each class is collected. Furthermore, The presented method can be easily applied in a semi-supervised way on a cohort of pathology cases to get novel segmentations, which would yield most likely correct segmentations in healthy tissue areas but would have many errors near pathology or artifacts. Hence, expert observers could subsequently refine the segmentations for use as a novel reference standard, which in turn can be used to train and improve the model to reliably and verifiably deal with pathology as well. We intend to address these issues in future work by adding more annotated patient scans to the dataset using the method described above, which will include pathology and foreign objects and will be from different scanners and acquisition protocols.

Our model has a straightforward design expecting a fixed number of timepoints, which works well for data from standardized acquisition protocols. However, dealing with a variable number of timepoints might be desirable in some cases. In this case, interpolation could be used, or recurrent layers could be used. Our model can be expanded with additional layers, filters, and shortcut connections to enable learning a richer set of problem-relevant feature maps. However, this remains technically challenging due to the total GPU RAM memory requirements for these experiments. Although these changes may improve the stability of the training process, it was not possible to increase the batch size to more than 2 or to increase our input spatial dimensions (picking a larger value for n, Figure 2.1) much further without altering the network. In the future, we would like to distribute our computations across multiple GPUs to cope with the memory requirements and scale to larger networks, which might involve switching to other deep learning frameworks.

Many aspects of the network architecture contributed to a successful deep learning model, like the number of multi-resolution levels, the number of feature maps, and the size of the filter kernels. our model has three max-pooling and upscaling operations, which provide feature extraction at four different resolution levels. The number of resolution levels can be changed by removing or adding a pair of max-pooling and upscaling operations and a long shortcut connection at a particular resolution. Generally, more resolution levels result in a bigger receptive field and hence each voxel can infer its class from a broader volume of surrounding context, but the minimal required input size increases. For example, increasing the resolution level from four to five results in an increase of the minimum required input subvolume size from 92^3 to 188^3, which would also increase the memory requirement by more than a factor of eight and would no longer fit in GPU RAM.

The number of feature maps could only be slightly increased due to memory limitations, but early experiments did not give a significant performance increase. In-

creasing the feature maps at the earlier layers is especially troublesome since the resulting intermediate feature maps take a lot of GPU memory. This problem is less expressed at the lower resolution layers, where each feature map uses approximately eight times less memory than a feature map at a previous resolution level. The minimal required feature maps for achieving similar performance were not investigated due to the required computation times, but it is expected that reducing the number of feature maps will at some point have a big effect on the performance.

The size of the filter kernels can be varied, but can be difficult to optimize, since it holds a close relation to the receptive field and therefore also the minimum input size for the network. Increasing all filter sizes from 3^3 to 5^3 for example requires much larger input subvolumes, which would not fit in GPU RAM anymore. Another approach would be to replace every pair of 3^3 filters by a single 5^5 filter, which effectively leaves the receptive field the same and reduces intermediate feature map computations at the cost of an extra non-linearity. Doing this properly involved lowering the initial learning rate.

Setting the training hyperparameters – like batch size, input size, optimizer choice, and optimizer parameters – was found to be at least as important as the network architecture to achieve good model performance. In our experience, changing the batch size and input size, had a great impact on training and final model performance. Generally taking the batch size and input size as big as possible while still being able to fit in GPU RAM memory worked best. For the optimizer, we have only experimented with default stochastic gradient descent with momentum. We did not test with other optimizers, but they might require some tweaking. We do not expect big performance increases from using different optimizers. Tweaking of the optimizer parameters, like the learning rate and momentum factor in our experience can have a big impact on training and final model performance.

Whereas predicting with our model is relatively fast (approximately 5 minutes for a full 4D-CTA case), the end-to-end training of the network could take several days. Hence, only a limited amount of experiments could be performed for this study. We parallelized our experiments over multiple Titan X GPUs to speed up training. Additionally, we split our training and validation computations per experiment and distributed these over different GPUs. We simultaneously used the CPU to prepare subvolumes while training on another subvolume on the GPU, to ensure the best possible continuity of input data. Furthermore, for validation we increased the input subvolume size to better utilize the GPU memory and predict slightly larger subvolumes, which also sped up the process significantly. It might be possible to further reduce training times through deep supervision approaches[66], by reducing some of the complexity of the model or by implementing more efficient data sampling schemes.

There is not much literature on CNNs with respect to handling 4D or higher dimensional data; yet, it is the opinion of the authors that deep learning approaches that are able to cope with high dimensional data will become increasingly important as datasets increase in size and incorporate more dimensions. Hence, the competitive segmentation results achieved by our proposed method, which was directly learned from 4D-CTA input, suggests potential application of the method beyond the application of stroke imaging.

3

Deriving 3D non-contrast CT from 4D-CTA

S.C. van de Leemput, M. Prokop, B. van Ginneken, R. Manniesing

Original title: Stacked Bidirectional Convolutional LSTMs for Deriving 3D Non-Contrast CT from Spatiotemporal 4D CT

Published in: IEEE Transactions on Medical Imaging, 39(4):985-996, 2019

Abstract

The imaging workup in acute stroke can be simplified by deriving non-contrast CT (NCCT) from 4D CT angiography (4D-CTA) images. This results in reduced workup time and radiation dose. To achieve this, we present a stacked bidirectional convolutional LSTM (C-LSTM) network to predict 3D volumes from 4D spatiotemporal data. Several parameterizations of the C-LSTM network were trained on a set of 17 4D-CTA/NCCT pairs to learn to derive a NCCT from 4D-CTA and were subsequently quantitatively evaluated on a separate cohort of 16 cases. The results show that the C-LSTM network clearly outperforms the baseline and competitive convolutional neural network methods. We show good scalability and performance of the method by continued training and testing on an independent dataset which includes pathology of 80 and 83 4D-CTA/NCCT pairs, respectively. C-LSTM is, therefore, a promising general deep-learning approach to learn from high-dimensional spatiotemporal medical images.

3.1 Introduction

Computed tomography (CT) is the preferred modality in the imaging workup of patients suspected of acute stroke since fast diagnosis is critical for patient outcome. A stroke workup consists of a non-contrast CT (NCCT) scan to identify hemorrhages, is followed by a CT angiography (CTA) to assess the blood flow within the cerebral vasculature, and is often followed by a 4D CT angiography (4D-CTA) to differentiate core (irreversibly damaged brain tissue) and penumbra (salvageable tissue)[67]. The CTA and 4D-CTA are respectively a 3D and 4D (sequence of 3D images) acquisition which are both acquired after the injection of contrast agent.

Randomized control trials published in 2015 including MR CLEAN[68] and others[69–72] have shown the benefit of endovascular therapy in ischemic stroke patients with proximal occlusions and have led to the inclusion of CTA imaging to the stroke guidelines[24] with the highest level of recommendation. Three recent randomized control trials (DAWN[73], DEFUSE 3[74], and EXTEND[75]) have unequivocally shown the value of 4D-CTA for patient selection beyond the recommended time window of six hours who will benefit from endovascular thrombectomy. These findings have led to the adoption of 4D-CTA imaging as the highest recommendation in the modern stroke guidelines[67]. Hence, 4D-CTA, just like CTA, is likely to become part of the clinical routine in the acute stroke imaging workup.

Ischemic stroke is the most prevalent type within acute stroke patients (87%[76]) which requires taking at least a CTA and often a 4D-CTA besides the conventional NCCT scan. When a patient enters the hospital with suspicion of stroke, a simplification of the stroke workup can be achieved by only acquiring a 4D-CTA and subsequently deriving the NCCT and CTA from the 4D-CTA, hereby reducing workup time, contrast usage, and radiation dose. The radiation doses are approximately in the ranges of 2.0-2.7, 2.8-5.4, and 5.0-6.0 mSv for respectively NCCT, CTA, and 4D-CTA[77–80]. The workup time is in the order of one to two minutes for each of the scans. The contrast usage for CTA and 4D-CTA are similar. Deriving the CTA and NCCT from the 4D-CTA is feasible because the 4D-CTA in principle contains more information than the other two scans. The feasibility of deriving high-quality CTA from 4D-CTA was shown in previous work[23].

We present a novel convolutional LSTM (C-LSTM) neural network designed to derive 3D volumes from 4D spatiotemporal data. The main contribution of this work is that we show the potential of the C-LSTM for deriving 3D volumes from 4D spatiotemporal data and we present the first application for deriving NCCT from 4D-CTA, which has the potential to simplify the stroke imaging workup.

3.1.1 Related work

C-LSTM[81] is a type of recurrent neural network which combines the long short-term memory (LSTM) network[82] – the standard for processing sequential data – with convolution neural networks[83] – the standard for processing spatial data – by replacing the internal matrix multiplications of the weights with the input and hidden states with convolutional operations. The added convolutions allow to simultaneously encode spatial features while also encoding long-term recurrent dependencies. In contrast, normal LSTMs can encode changes on the pixel level, yet they cannot encode spatial features over time (e.g., motion). Hence, C-LSTM networks are different from methods stacking normal LSTM networks on top of conventional convolutional layers, although these are often found under the same name in the literature. We will only consider C-LSTM networks that have convolutions integrated for the remainder of this paper.

The C-LSTM model has been first introduced in[81] to predict the weather from video sequences. The model was designed to encode spatiotemporal information in general, hence the model has found its application in a range of domains: video analysis[84,85], human pose and gesture recognition[86–88], various sensor array monitoring setups[89–92], and protein structure prediction[93,94]. There have been a few applications of the C-LSTM model for medical image data, but most focus on segmentation tasks and predicting a 3D volume from a sequence of 2D slice-based (2D + slice) instead of predicting a 3D volume from temporal sequences of 3D volumes (3D + time)[95–97].

Despite the many interesting applications of the C-LSTM, it has not yet been applied to derive 3D volumes from 4D spatiotemporal medical images or 4D-CTA data. Furthermore, most C-LSTM applications have been limited to 3D spatiotemporal video data (2D + time) and were not designed to handle 4D dynamic volumetric data (3D + time).

Only a few deep learning approaches have been presented that were applied to 4D medical data when excluding multi-modal/multi-channel 3D data. Shin et al.[50] used stacked autoencoders for unsupervised multiple organ detection in dynamic MRI data, but produced rough segmentations at best. Xu et al.[98] automatically aligned and analyzed convergent beam electron diffraction patterns from big 4D micro electroscopic data using a hierarchy of several classical feed-forward convolutional neural networks (CNN). The only other existing application of deep learning on 4D-CTA data within the literature is experimental work by Vargas et al.[99], yet they focused on classification, and their evaluation was somewhat limited. None of the mentioned works use deep learning and 4D-CTA data.

Several CNNs for medical image derivation and reconstruction exist in the liter-

ature reporting overall improved performance over traditional approaches (see[100] and[101] for an overview of these topics). The CNN methods typically employ a regression approach, i.e., optimizing the L2 loss between target and derivation. Nie et al.[102] derived CT from MRI images using four 3D convolutional layers. Bahrami et al.[103] used a CNN network for deriving 7T from 3T MRI with four 3D convolutional layers. Others used CNNs to perform image denoising on low-dose 2D CT images to derive denoised images[104,105]. However, the majority of the proposed regression approaches only cover 2D or 3D images and are not designed to account for the temporal information of the 4D-CTA.

Generative adversarial networks (GAN)[106] are increasingly used for image generation and derivation[107,108]. In this setting, two networks are trained in competition: a generator that tries to generate images looking similar to the target image distribution, and a discriminator that tries to distinguish between images made by the generator and the real images. A well-trained generator can create images that appear to be drawn from the real target data distribution. A popular GAN for image synthesis is the CycleGAN[109], which can synthesize images when trained with unpaired target and source images. This technique has been applied in medical imaging[110] for 2D unpaired MR to CT synthesis. However, GANs have not yet been applied to 4D data nor NCCT derived from 4D-CTA data, and these methods are more demanding with respect to memory usage and processing time. We preferred a simple regression training scheme over adversarial training: The generator network within a GAN training scheme is known to mimic the source data distribution[106], which contains a lot of undesired noise in the use case we considered in this work. Minimizing the mean squared error between derivation and target has a simpler training procedure, is simpler to optimize during model training, and yields smoother results[110].

We introduce a general stacked bidirectional convolutional LSTM deep learning model for deriving 3D volumes from spatiotemporal 4D data. Using this model, the feasibility of deriving a NCCT from 4D-CTA is demonstrated. This is a major extension of our previous work: the stacked C-LSTM model was simplified by removing the extra 3D convolutions on top of the C-LSTM layers, and ablation experiments were added for the hyperparameters and for the number of timepoints[111]. Additionally, the model was compared to state-of-the-art deep learning methods and validated for scalability to a larger dataset and cases with pathology.

3.2 Methods

3.2.1 C-LSTM

The convolutional LSTM model (C-LSTM) from[81] was used, which is an adaptation of the normal LSTM model and can be described by the following equations:

$$
\begin{aligned}
i_t &= \sigma(x_t *_x W_{xi} + h_{t-1} *_h W_{hi} + b_i) \\
f_t &= \sigma(x_t *_x W_{xf} + h_{t-1} *_h W_{hf} + b_f) \\
o_t &= \sigma(x_t *_x W_{xo} + h_{t-1} *_h W_{ho} + b_o) \\
g_t &= \phi(x_t *_x W_{xc} + h_{t-1} *_h W_{hc} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \omega(c_t)
\end{aligned}
\tag{3.1}
$$

where x_t and h_{t-1} are the inputs at timepoint t, with x_t the input sequence data at timepoint t and h_{t-1} the previous hidden state. h_t is the output at timepoint t and also the hidden input state for the next timepoint $t+1$. i_t, f_t, o_t, and g_t are the input gate, the forget gate, the output gate, and the cell state, respectively; this encodes how much the input at the current timepoint and the hidden state from the previous timepoint contribute to the current cell state c_t through the weight matrices W_x and W_h and biases b. Usually, σ and ϕ are the sigmoid and hyperbolic tangent functions. $\odot$ is the element-wise product and $*_x$ and $*_h$ are the convolution operators for the input and the recurrent input respectively. ω was set to the hyperbolic tangent function. Figure 3.1 shows a graphical representation of the C-LSTM equations for sequentially processing 3D spatiotemporal data.

Note that the C-LSTM is essentially a generalization of the conventional LSTM. The normal LSTM computations can be obtained for the C-LSTM model by setting a convolutional kernel of 1^3 for $*_x$ and $*_h$ or by entirely replacing the convolution operations with matrix multiplications.

C-LSTM layer

Since the C-LSTM falls within the class of recurrent neural networks, it can have any of the following input-output sequence mappings: one-to-one, one-to-many, many-to-one, or many-to-many. However, we only considered two variants which encapsulate Eqs. (3.1) in a single layer. This results in a function $F : S \to S$ which takes in a sequence S of length l and outputs an equally lengthy sequence $(h_1, h_2, \ldots, h_{l-1}, h_l)$ or, alternatively, only the last state in the sequence h_l.

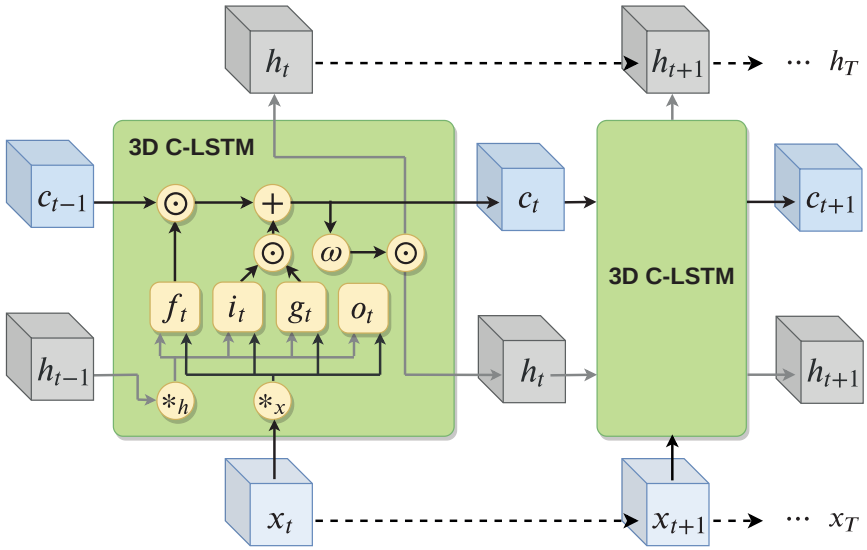


Figure 3.1: Graphical representation of a single 3D convolutional long short term memory (3D C-LSTM) layer for processing 4D spatiotemporal data. Each cube represents a 3D subvolume. Input 4D spatiotemporal sequence $(x_0, x_1, \ldots, x_{T-1}, x_T)$ of length T with each 3D subvolume x_t at timepoint t can be found at the bottom. Each input subvolume x_t with the hidden state h_{t-1} and cell state c_{t-1} subvolumes from the previous timepoint (on the left) are fed into the C-LSTM equations 3.1 resulting in a new cell state subvolume c_t and a hidden/output subvolume h_t (on the right). By repeatedly feeding the subvolumes x_t from the sequence to the C-LSTM equations and combining the resulting hidden/output states h_t you obtain a novel processed 4D spatiotemporal sequence $(h_0, h_1, \ldots, h_{T-1}, h_T)$ of length T (at the top).

Bidirectional C-LSTM

In a bidirectional approach to sequences, the signal is processed both from 0 to N and also from N to 0 by another similar recurrent network. Finally, the two results are combined, and this generally yields better results. Since we knew the entire sequence length beforehand, the bidirectional approach was used and the sequences at the end were summed; given sequence output 1 $(h_1^1, h_2^1, \ldots, h_{l-1}^1, h_l^1)$ and the reversed but same-sized sequence output 2 $(h_l^2, h_{l-1}^2, \ldots, h_2^2, h_1^2)$, the output sequence y of the combined bidirectional LSTM becomes $y = (h_1^1 + h_l^2, h_2^1 + h_{l-1}^2, \ldots, h_{l-1}^1 + h_2^2, h_l^1 + h_1^2)$.

Stacked C-LSTM

The previously described components were combined in a network consisting of a parameterizable stack of C-LSTM layers and convolutions. A schematic overview of the $K, f, *_x, *_h$-stacked C-LSTM network is shown in Fig. 3.2.

The network takes a batch of one or more 3D spatiotemporal input sequences of length T timepoints as a 6D tensor with the following dimensions: batch size, timepoints, number of filters, and the spatial dimensions (z, y, x). The filter dimension was introduced for implementation convenience following the Keras data format and was always set to one for the input tensor. The batch of sequences was fed through a stack of K bidirectional C-LSTM layers, each with f filters, $*_x$ input convolution kernel size, and $*_h$ hidden convolution kernel size. All K bidirectional C-LSTM layers passed on the entire length of the sequences to the next layer, which was the entire output sequence y composed from the forward and backward hidden states h^1 and h^2 as described in section 3.2.1. The last layer in the stack only passed a single prediction (y_T) for its input sequence of length T, reducing the input sequences to a 5D tensor with a filter size of f.

Lastly, a final single 3D convolution with a single filter, a 1^3 convolutional kernel, and an identity activation function was used to produce the output-derived image as a 5D tensor.

3.2.2 Model training

The training of the model employed a regression training scheme where the mean squared error loss (MSE) between the NCCT and derived NCCT was minimized using the RMSProp optimizer. The RMSProp optimizer was chosen because initial experiments yielded more stable training performances than the SGD optimizer. The optimizer settings were a learning rate of 0.001, a rho value of 0.9, and an epsilon of 1e-6. Each model was trained for 1500 iterations, where each iteration consisted of 100 randomly sampled 4D-CTA sub-volumes ($32^3 \times 19$ timepoints) from within the cranial cavity across all training set cases. The choice for the sub-volume size was based on the large memory requirements of the model on the GPU during training. Methods for reducing the memory footprint, like reducing batch size, gave a worse performance. A cranial cavity mask was created to segment all intracranial soft tissue using the method of Patel et al.[18]. The resulting cranial cavity mask was refined by discarding all voxels with an intensity below air density (-1000 Hounsfield units) followed by a binary erosion with a 3D ball structuring element with a radius of three voxels. The batch size during training was 2. Training and evaluation were performed on a single NVIDIA Titan X GPU with 12 GB of RAM using Theano[58] as the backend.

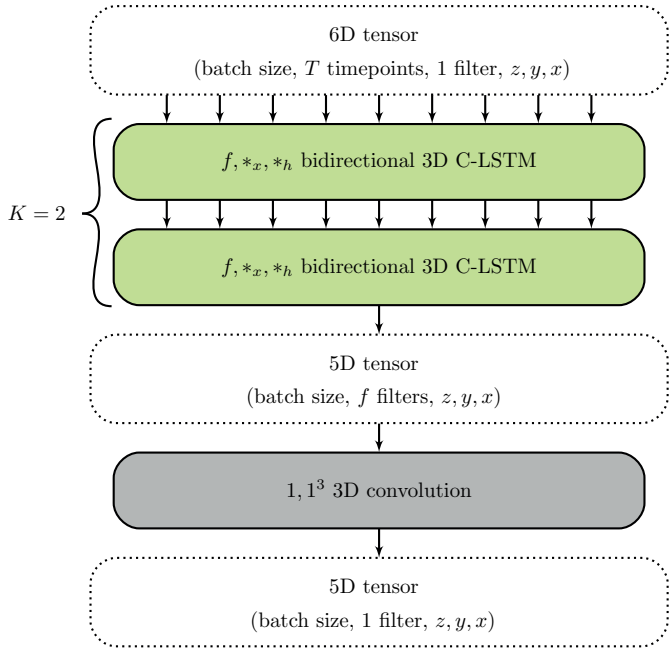


Figure 3.2: Parameterizable $K, f, *_x, *_h$-stacked C-LSTM network. In green, K bidirectional C-LSTM layers with f filters, $*_x$ the size of the input kernel, and $*_h$ the size of the recurrent kernel. In gray, a single 3D convolution with one filter, 1^3 kernel, and identity activation function. At the top, the input sequence is a 6D tensor (a number of 3D subvolume sequences of T timepoints, with a single dummy filter) which is reduced to a 5D tensor with f filters after the last C-LSTM layer. Subsequently, the single filter 3D convolution renders the final output a single filter 5D tensor at the bottom.

The initial hidden state of each C-LSTM layer was set to zero. The weights W_{xi}, W_{xf}, W_{xo}, and W_{xc} were all initialized using uniform Xavier initialization[112]. The recurrent kernels W_{hi}, W_{hf}, W_{ho}, and W_{hc} were all initialized using random orthogonal matrices. All bias terms were set to zero except for the forget bias, which was set to one as recommended by Jozefowicz et al.[113]. All normal convolutional layers were initialized using uniform Xavier initialization.

3.2.3 Implementation details

The stacked 3D C-LSTM models were implemented in Keras[114]. The C-LSTM operations at each timepoint were optimized by exploiting that i_t, f_t, o_t, and g_t from equation 3.1 require similar computations. Hence, the convolution operations $*_h$ and $*_x$

can be computed efficiently by concatenating the weight matrices for W_x and W_h, i.e., $x_t *_x (W_{xi}, W_{xf}, W_{xo}, W_{xc})$ and $h_{t-1} *_h (W_{hi}, W_{hf}, W_{ho}, W_{hc})$. This way, the components for i_t, f_t, o_t, g_t could be computed by two convolutions instead of eight.

3.3 Data

This retrospective study included 196 patients (age 65 ± 13 years, 59% male) with suspicion of stroke admitted to our hospital in 2015-2017 and who have received both a NCCT and a 4D-CTA scan. 63 cases were diagnosed with at least one major pathology (e.g., hemorrhage, large infarct, and ischemic symptom) and 30 cases showed at least one major artifact (e.g., clips, streaking artifacts, and metal artifacts). Two small datasets were taken from the total patient data for tuning the hyperparameters: 17 cases for training $\mathbb{D}_A^{train}$ and 16 cases for testing $\mathbb{D}_A^{test}$, with 8 and 0 pathology cases respectively. The remaining data was added to a larger disjoint dataset of 163 cases consisting of a training set $\mathbb{D}_B^{train}$ of 80 (28 pathology and 14 artifact) cases and a test set $\mathbb{D}_B^{test}$ of 83 (27 pathology and 16 artifact) cases.

4D-CTAs were acquired on a 320-row CT scanner (Toshiba Aquilion ONE, Japan) consisting of 19 volumetric scans with different exposures per timepoint. Patients received 80 mL of contrast agent (Iomeron) injected in the cephalic vein at the start of the first acquisition. Image reconstruction was done using an FC41 smooth convolution kernel, resulting in $512 \times 512 \times 320$ voxels with a voxel size of $0.47 \times 0.47 \times 0.5$ mm. NCCTs were acquired on the same scanner reconstructed with an FC26 kernel yielding $512 \times 512 \times 302$ voxels with a voxel size of $0.43 \times 0.43 \times 0.5$ mm.

3.3.1 Preprocessing

All 4D-CTA timepoints $t > 0$ were rigidly registered to the first 4D-CTA timepoint ($t = 0$) to correct for potential head movement during acquisition. The registration was performed using the method and parameter settings as described by Manniesing et al.[30]. The NCCT was rigidly registered to the same space of the first timepoint of the 4D-CTA with Elastix[60] using similar settings. This registration step also resulted in the same resolution for the NCCT with respect to the 4D-CTA. Finally, before neural network training and prediction, each voxel HU value x was linearly scaled by $f(x) = (x + 50)/250$. This operation was reversed after training and prediction to map it back to HU.

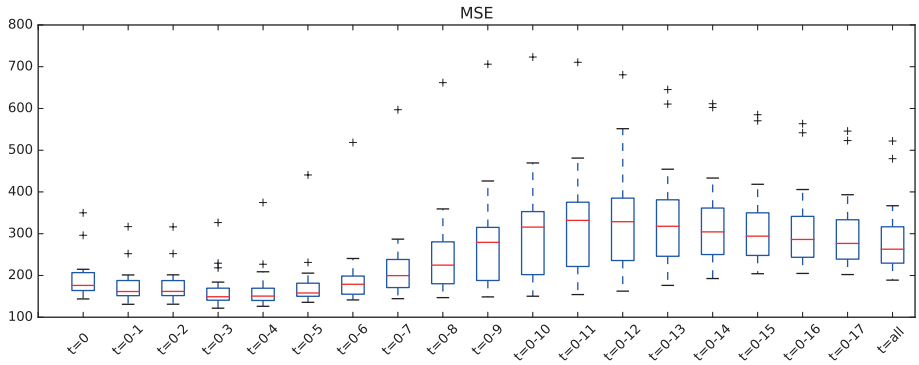


Figure 3.3: Shows mean squared error (MSE) performance for the first $t = 0\text{-}n$ timepoint averages of a 4D CT angiography image as a derivation for the non-contrast CT (NCCT) target. This plot indicates that combining the first four timepoints (t=0-3) approximates the NCCT derivation the best among the other timepoint averages.

3.4 Evaluation

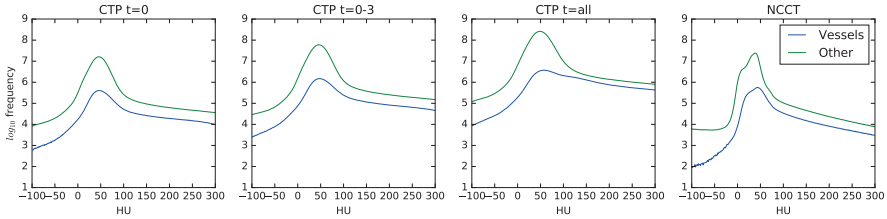
3.4.1 Quantitative evaluation

All methods were compared using the following regression error measures: mean squared error (MSE), r^2 score, and structural similarity index (SSIM)[115]. However, the diagnostic relevant information of a NCCT is only found within the cranial cavity. Hence, only the voxels within the cranial cavity mask (described in section 3.3.1) were used for computing these quantitative metrics.

To test for statistically significant differences between methods, the MSE, r^2 scores, and SSIM were first computed for all test set cases. Next, for each metric and method, the normality of the scores was estimated using the Shapiro-Wilk test for normality. If two competing methods were normally distributed, a subsequent paired $t-$test was used to assess a significant difference between them, otherwise, a Wilcoxon signed-rank test was used instead. The significance level was set to .05.

3.4.2 Baseline derivation models

A lower bound baseline was established using three naive derivation methods from the 4D-CTA: taking the first timepoint (t=0), taking the mean of the first three timepoints (t=0-3), and taking the average over all timepoints (t=all). Taking the first timepoint is an obvious approach for deriving a NCCT, since it is the 4D-CTA timepoint with the highest exposure and hence has the best signal-to-noise ratio. Also,



tissue class	vol. %	4D-CTA t=0	4D-CTA t=0-3	4D-CTA t=all	NCCT
vessel	3.8%	73.7 HU	76.2 HU	110.1 HU	54.1 HU
non-vessel	96.2%	49.0 HU	49.3 HU	53.1 HU	35.2 HU
combined	100.0%	50.0 HU	50.3 HU	55.2 HU	35.9 HU

Figure 3.4: Intensity histograms of Hounsfield units (HU) within the $[-100, 300]$ domain averaged over the entire dataset of 39 patients. The dataset was examined for different timepoint partitions within each 4D CT angiography (4D-CTA) (t=0, t=0-3, and t=all) and the non-contrast CT (NCCT) for vessel and non-vessel tissues within the cranial cavity. The table shows the average HU for the 4D-CTA partitions and NCCT images.

the contrast agent is less expressed earlier in the time sequence, which is closer to the signal intensity of the target NCCT. Thus, averaging the first few timepoints reduces derivation noise within individual timepoints while avoiding the contrast signal from the later scans. Furthermore, t=all was used as a reference for having the optimal noise reduction over timepoints, but this derivation includes a signal from the contrast material present at later timepoints. Figure 3.3 shows the average MSE performance by using the first n timepoints of the 4D-CTA as a derived NCCT, which exposes t=0-3 as the optimal baseline.

An analysis of the cranial cavity intensity histograms over all patients within the domain of $[-100, 300]$ HU on the 4D-CTA and NCCT shows a basic intensity bias between the two imaging types. The intensity histograms can be found in Fig. 3.4 and are divided into vessel and non-vessel tissue type counts. All voxels with values > 110 from the temporal variance of the 4D-CTA were assigned to the vessel class.

This bias must be accounted for when computing the quantitative evaluation metrics to avoid underestimating the derived image quality of these models. The bias was estimated as the difference in average non-vessel tissue intensity of the NCCT and 4D-CTA classes: -13.9, -14.1, and -17.9 HU for t=0, t=0-3, and t=all, respectively. The learned models in this work did not suffer from this bias because these methods learned to estimate this bias from the data.

3.5 Experiments

Four different types of experiments were performed in this study. The first set of experiments assessed the best hyperparameter configuration of the network using ablation experiments. The second set of experiments compared the best parameterization of the stacked C-LSTM architecture with several naive baselines and competitive CNN methods from the literature. In the third set of experiments, the best model was used to investigate the trade-off between the number of timepoints used for the derivation and the derived image quality. In the final set of experiments, the model was further optimized and evaluated on a larger dataset with pathology cases to test the scalability of the model.

3.5.1 Ablation experiments

A series of nine ablation experiments were performed using the parameterized stacked C-LSTM architecture to investigate the contribution of the individual hyperparameters: input convolution kernel size $*_x$, hidden convolution kernel size $*_h$, and the number of C-LSTM stacks K. In the first six experiments, the optimal choice for kernel sizes $*_x$ and $*_h$ were determined by fixing the filter size f to 64 and the number of C-LSTM stacks K to one and varying the kernel sizes $*_x$ and $*_h$. In the final three experiments, the optimal number of C-LSTM stacks was determined by fixing $*_x$ and $*_h$ to the previous optimal setting for $K = 1$ while varying the number of stacks $K \in \{2, 3, 4\}$. All experiments used bidirectional C-LSTM stacks.

Each parameterization of the stacked C-LSTM architecture was trained from scratch for 1500 iterations using dataset $\mathbb{D}_A^{train}$ from section 3.3. The metrics were computed by averaging each metric (MSE, r^2, and SSIM) within the brain mask over all test set cases $\mathbb{D}_A^{test}$. Since computing the test-set scores was very time-consuming, the network was evaluated on the test-set of 16 cases only once every 250 iterations; the best-performing iteration on the MSE was selected as the final model score. The ablation experiments with their parameterizations and final scores can be found in Table 3.1.

The first six experiments were used to determine the best hyperparameters for the subsequent experiments. From these first experiments, the network with the highest average metrics on r^2 and SSIM and the lowest average metrics on MSE was selected if it was significantly better than the others. To determine the final best hyperparameter settings, the same procedure was used but for all nine experiments.

3.5.2 Baseline comparison

To position the performance of the best-trained parameterization of the stacked C-LSTM architecture (obtained in the ablation experiments from section 3.5.1) was compared with two state-of-the-art estimation methods (a 3D U-Net[40] and the method of Nie et al.[102]) and the naive baselines t=0, t=0-3, and t=all (section 3.4.2). All methods were applied to the same test set of dataset $\mathbb{D}_A$ (see section 3.3) and evaluated on MSE, r^2, and SSIM. The 3D U-Net and the method of Nie et al. were trained on dataset $\mathbb{D}_A^{train}$ for 1500 iterations each, closely matching the training for the C-LSTM. Similarly, the best-performing models for the methods were estimated by picking the best-scoring model at every 250th iteration on the test set $\mathbb{D}_A^{test}$.

The method of Nie et al.[102] is a four-layer deep 3D convolutional network with four 3D-convolutions with kernel sizes of 7^3, 5^3, 3^3, and 3^3 and filter sizes of 32, 64, 32, and 1, respectively. The network was trained using the RMSProp optimizer while optimizing the MSE. The data was presented to the network in batches of two samples, with each sample having the 19 4D-CTA timepoints encoded as channels with 128^3 voxels. The model was trained for 1500 iterations processing 50 batches per iteration.

The 3D U-Net[40] is a well-known model in medical imaging. The model has two pathways: a downward analysis path – which analyses the data at various resolutions by applying several two convolution layers followed by a single max pooling operations – and an upward synthesis path – which reintegrates the lower resolution information from the downward path to the original high-resolution output using deconvolution and pairs of convolution operations. The 3D U-Net is typically applied to image segmentation, but by removing the final softmax operation it can serve as a multiscale regression model as well.

For this work, we took a 3D U-Net with 3 max-pooling and 3 corresponding upscale operations. For upscaling nearest neighbor, upsampling was used instead of deconvolution. An initialization scheme by He et al.[57] was used to initialize the weights. The model was trained to minimize the MSE using the RMSProp optimizer for 1500 iterations. During one iteration, 100 batches were processed consisting of a single cropped 4D-CTA sequence of 19 data volumes of $116 \times 132 \times 132$ voxels each.

3.5.3 Timepoint ablation experiments

One of the strengths of recurrent networks is that they can deal with sequences of varying lengths. This applies to CT, since decreasing the number of timepoints would decrease radiation exposure to the patient while providing valuable information for optimizing the 4D-CTA acquisition protocol. In the following set of experiments, we

utilized the best C-LSTM model found in the ablation experiments (section 3.5.1) to estimate the trade-off of the derived image quality by ablating timepoints.

We evaluated our pre-trained model by applying it on inputs from the 16 test set cases from dataset $\mathbb{D}_A^{test}$ with varying sequence lengths t=0-n, always starting with the first timepoint t=0 up to t=n where $n \in \{0, 1, \ldots, 18\}$; note that t=0-18 is equivalent to including all timepoints (t=all). Next, the derived images were scored on MSE and EV. Since the model was not optimized for handling varying timepoints during initial training, additional training on the training set of $\mathbb{D}_A^{train}$ for each of the corresponding input sequence lengths was performed for 0, 100, 200, and 300 additional training iterations.

3.5.4 Assessing scalability

The scalability of the method to larger datasets and cases with major pathology and artifacts was assessed on a completely separate dataset $\mathbb{D}_B^{train}$ of 80 training cases and 83 test cases $\mathbb{D}_B^{test}$. The best C-LSTM model, 3D U-Net, and the method of Nie et al. were each trained for an additional 1500 iterations on the bigger training set, and evaluated on a separate validation set of 16 cases at every 250th iteration. For each of the three model types, the model scoring the best results based on the average MSE, r^2, and SSIM on the validation set $\mathbb{D}_A^{test}$ from all evaluated iterations was evaluated a final time on the full test set. The baseline method results were also evaluated on the full test set $\mathbb{D}_B^{test}$ for comparison.

3.6 Results

3.6.1 Ablation experiments

Table 3.1 shows the results on the test set $\mathbb{D}_A^{test}$ (16 cases) of the nine ablation experiments with the parameterized stacked C-LSTM models after completing training. Within the six different kernel size parameterizations (exp. 1-6), experiments 3 and 5 significantly outperformed all the other methods on all the evaluated metrics ($p < 0.05$). There were no significant differences between experiments 3 and 5 ($p > 0.05$), therefore the simplest- and cheapest-to-compute kernel parameterization from experiment 3 with $*_x = 3^3$ and $*_h = 3^3$ was used for the final three experiments (7-9). The trained stacked C-LSTM model from experiment 7 with $K = 2$ significantly outperformed all other experiments with $p < 0.05$ on all performance metrics.

Wall clock training times for the ablation experiments were approximately 6-7 days for experiments 1-5, 13 days for experiments 6 and 7, and 18 and 23 days for exper-

Table 3.1: Ablation experiments with the C-LSTM hyperparameters. The first 1-6 experiments varied the convolution size parameters $*_x$ and $*_h$; experiments 7-9 varied the number of C-LSTM stacks K. The listed scores are the best average performances on the test set after 1500 iterations of training on mean squared error (MSE), r^2, and structural similarity index (SSIM). Parameter $f = 64$ was fixed for all experiments.

exp.	$*_x$	$*_h$	K	MSE	r^2	SSIM
1	1^3	1^3	1	90.84	0.500	0.292
2	3^3	1^3	1	88.40	0.511	0.286
3	3^3	3^3	1	75.04	0.590	0.332
4	5^3	1^3	1	88.60	0.511	0.283
5	5^3	3^3	1	76.99	0.578	0.327
6	5^3	5^3	1	93.37	0.486	0.239
7	3^3	3^3	2	70.88	0.611	0.356
8	3^3	3^3	3	78.15	0.570	0.291
9	3^3	3^3	4	87.27	0.519	0.285

Table 3.2: Baseline comparison of the best parameterization of the C-LSTM against a trained 3D U-Net, a trained model based on Nie et al., and the baseline derivation models (t=0, t=0-3. t=all). The listed scores were the best average performances on the test set after 1500 iterations of training by mean squared error (MSE), r^2, and structured similarity index (SSIM). The best C-LSTM model from experiment 7 (Table 3.1) was used.

exp.	MSE	r^2	SSIM
c-lstm	70.88 ± 20.73	0.61 ± 0.09	0.356 ± 0.06
u-net	105.03 ± 22.40	0.43 ± 0.06	0.268 ± 0.04
nie	191.95 ± 39.96	-0.04 ± 0.05	0.064 ± 0.01
t=0	194.03 ± 53.87	-0.08 ± 0.28	0.263 ± 0.05
t=0-3	167.53 ± 50.53	0.07 ± 0.24	0.319 ± 0.06
t=all	288.68 ± 94.11	-0.63 ± 0.63	0.311 ± 0.05

iments 8 and 9, respectively. Evaluation wall clock times for the first 5 experiments varied from 7-16 mins/case, and the last 4 experiments took 36, 36, 56, and 65 mins/case, respectively.

3.6.2 Baseline comparison

Table 3.2 and Figure 3.5 show quantitative performance results between the best performing C-LSTM model, 3D U-Net, method of Nie et al., and the baselines: t=0, t=0-3, and t=all. The performances of all methods differed significantly from each other ($p < 0.05$), except for the method of Nie et al. and the t=0 baseline on MSE and r^2, and 3D U-Net and the t=0 baseline for the SSIM ($p > 0.05$). Also, the method of Nie et al. and the t=0-3 baseline performances did not differ significantly on r^2 ($p > 0.05$).

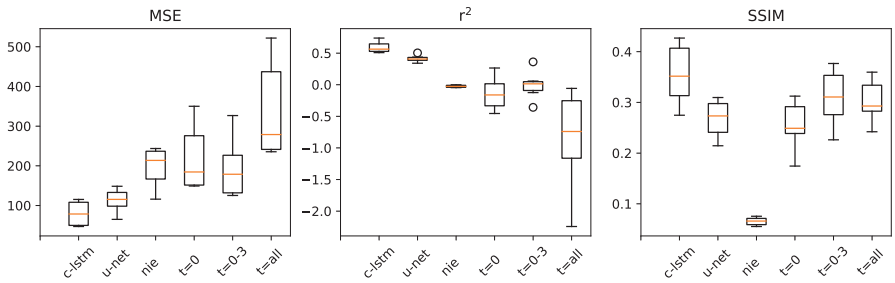


Figure 3.5: Final results on the test set $\mathbb{D}_A^{test}$ for all methods: our C-LSTM model, the best-trained 3D U-Net, the best-trained model of Nie et al., and the 3 baseline methods (t=0, t=0-3, t=all). Metrics from left to right: mean squared error (MSE), r^2, and structured similarity index (SSIM).

3.6.3 Timepoint ablation experiments

Figure 3.8 shows plots on the trade-off between removing timepoints from the input 4D-CTA and the derived image quality for the C-LSTM on MSE and SSIM. The upper bounds for both MSE and SSIM suggest a slight reduction in derived image quality after ablation of five timepoints at t=0-12. The line at 0 additional train iterations suggests that additional fine-tuning is necessary as the number of ablated timepoints increases, with an exponential decay in performance with the number of ablated timepoints.

Wall clock training times for the timepoint ablation experiments scaled linearly with the number of included timepoints T with the following function yielding the time in hours per 300 training iterations: $f_{\text{train}}(T) = 1.49 + 3.11T$. This yielded 4.59 hours for $T = 1$ up until 57.43 hours of training time for $T = 18$, for the 300 training iterations. Similarly, evaluation wall clock time scaled linearly in the number of processed timepoints T with the following formula given in minutes per case: $f_{\text{predict}}(T) = 4.64 + 1.14T$, from 5.78 mins/case for $T = 1$ to 25.22 mins/case for $T = 18$.

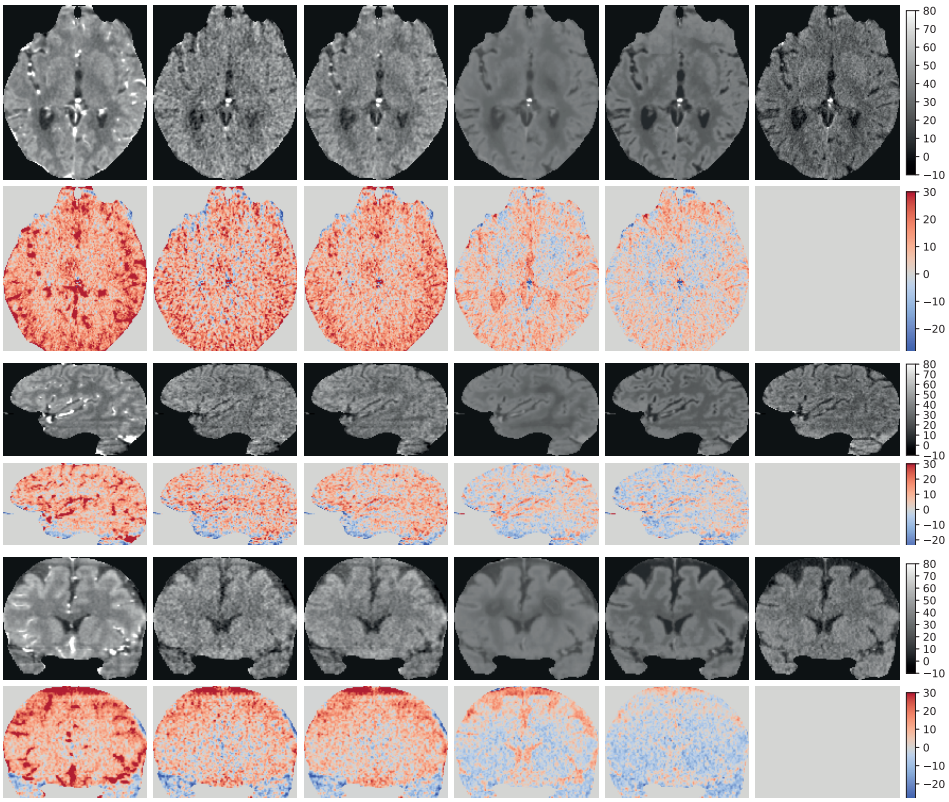


Figure 3.6: Qualitative results on three different cases from the test set $\mathbb{D}_B^{test}$ showing slices of the intracranial tissue. From top to bottom for every two rows: an axial, coronal, and sagittal slice. From left to right the following derivation models: t=all, t=0, t=0-3, best 3D U-Net, best C-LSTM model from experiment 1, and the target reference non-contrast CT (NCCT). Every second row depicts the difference image between reference NCCT and the derived image from the row above it. The scales at the right are in Hounsfield units (HU).

3.6.4 Assessing scalability

The performance of the C-LSTM model, 3D U-Net, and method of Nie et al. trained on the larger dataset $\mathbb{D}_B^{train}$, and of the naive baselines t=0, t=0-3, and t=all evaluated on the separate larger test set $\mathbb{D}_B^{test}$ on the regression metrics can be found in Table 3.3. One of 83 test NCCT cases suffered from major streak artifacts and was a few millimeters off registration; this heavily skewed the results for all methods and metrics in the analysis, hence it was treated as an outlier and excluded. Figure 3.9 lists the same metrics as data points for each case, partitioned per normal-appearing

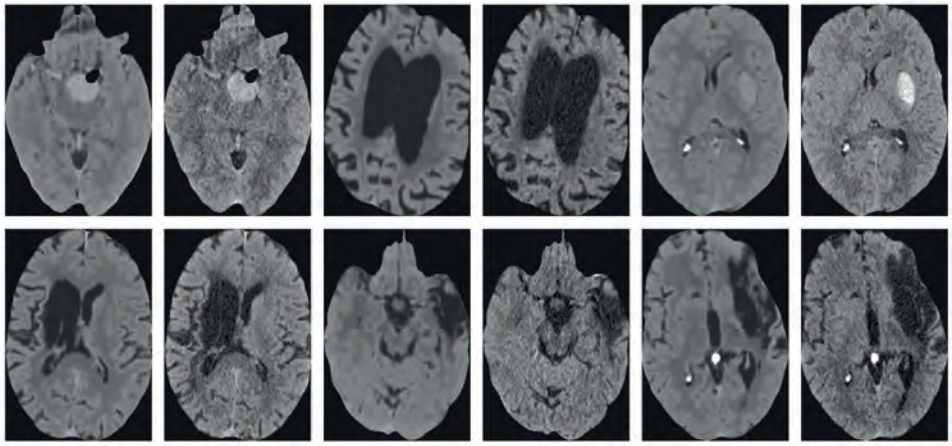


Figure 3.7: Qualitative assessment of six different patients with pathology from test set $\mathbb{D}_B^{test}$. For each patient, two images from the same axial slice are shown with on the left the best derived NCCT from the C-LSTM model and on the right the reference NCCT. The left and center pairs are examples of adequate derivations for the present pathology (hemorrhage, enlarged liquor, and parenchyma defects). In contrast, the derived quality of the right pairs can be improved. That is: the hemorrhage in the top right image should ideally be brighter to have higher contrast and for the bottom right image the delineation of the parenchyma defect at the border of the skull should have been more hypodense.

cases and cases with pathology. The best-performing C-LSTM model based on all metrics on the validation set was found to be the model at the last iteration 1500. Significant differences were found between the C-LSTM model and the baseline performances for all metrics within normal cases, pathology cases, and all cases combined ($p < 0.05$).

Qualitative visualizations of several slices of the derived NCCT images on test set $\mathbb{D}_B^{test}$, their respective reference NCCT, and the related approximation error can be found in Figure 3.6. Qualitative visualizations of pathology slices for the derived NCCT images for the best model and the reference NCCT images can be found in Figure 3.7.

Wall clock training times for training the C-LSTM was 15 days; evaluation was performed with 36 mins/case on average.

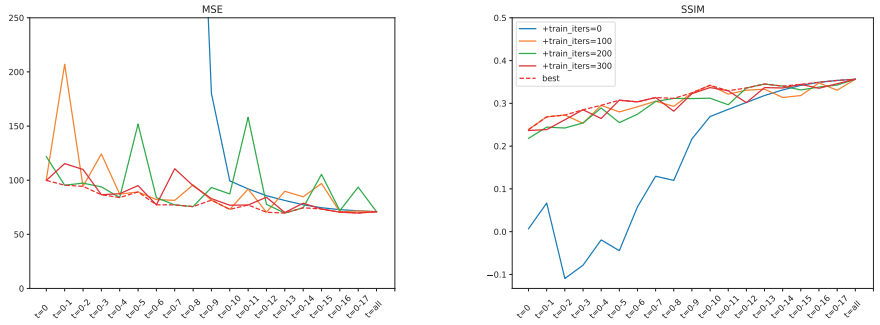


Figure 3.8: Assessment of the number of input timepoints on the C-LSTM model performance for target NCCT derivation. On the leftmost graph: the number of used timepoints t=0-n versus mean squared error (MSE). On the right graph: the number of used timepoints t=0-n versus structural similarity index (SSIM). Different lines indicate model performance after 0, 100, 200, and 300 training iterations starting with the best C-LSTM model trained on 19 timepoints. The red dotted line gives the best performance across all iterations.

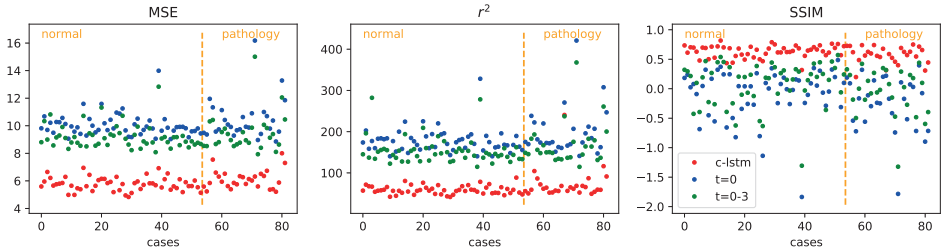


Figure 3.9: Score per case on the test set $\mathbb{D}_B^{test}$ (82 cases) for the following methods: our C-LSTM model (red), and two of the three baseline methods t=0 (green) and t=0-3 (blue). Metrics from left to right: mean squared error (MSE), r^2, and structural similarity index (SSIM). Each graph has been partitioned into normal cases (left) and pathology cases (right).

3.7 Discussion

We have presented a stacked bidirectional C-LSTM architecture for deriving 3D images from 4D spatiotemporal data. The model was able to derive NCCT images from 4D spatiotemporal 4D-CTA data with better performance on MSE, r^2, and SSIM than three baseline methods and two other state-of-the-art deep learning methods (see Table 3.2).

Figure 3.6 shows the ability of the C-LSTM model to encode both the spatial and the temporal information. The vessels were completely suppressed in the final model

Table 3.3: Comparison of the best C-LSTM parameterization against the baseline derivation models (t=0, t=0-3, t=all) and the two other state-of-the-art deep learning derivation methods (3D u-nUet and Nie). The listed scores are the best average performance on the test set after 1500 iterations of training on mean squared error (MSE), r^2, and structural similarity index (SSIM). One of 83 test cases was excluded because it was a severe outlier.

model	MSE	r^2	SSIM
normal			
c-lstm	58.74 ± 10.20	0.601 ± 0.12	0.361 ± 0.06
u-net	108.14 ± 40.49	0.286 ± 0.29	0.271 ± 0.04
nie	163.97 ± 45.36	-0.053 ± 0.06	0.062 ± 0.01
t=0	177.04 ± 29.12	-0.224 ± 0.45	0.255 ± 0.04
t=0-3	153.26 ± 31.56	-0.057 ± 0.39	0.309 ± 0.05
t=all	260.90 ± 62.04	-0.794 ± 0.67	0.294 ± 0.05
pathology			
c-lstm	72.83 ± 36.77	0.532 ± 0.15	0.330 ± 0.07
u-net	118.12 ± 54.29	0.244 ± 0.32	0.254 ± 0.05
nie	174.51 ± 53.44	-0.105 ± 0.10	0.063 ± 0.01
t=0	199.59 ± 58.02	-0.365 ± 0.61	0.235 ± 0.05
t=0-3	168.70 ± 50.68	-0.153 ± 0.53	0.289 ± 0.06
t=all	279.72 ± 78.01	-0.930 ± 0.85	0.281 ± 0.05

predictions, but the traces of calcification (small high-density spots on the NCCT and the derived NCCT seen in the top row image) were not. Comparing the temporal average of the 4D-CTA with the derived NCCT shows that the model was able to overcome the general intensity bias between the 4D-CTA and the NCCT target. The model also created better contrast of the cerebrospinal fluid at the giri and sulci with the brain tissue. Furthermore, the derived NCCT contained much less noise and produced a smoother result, which might be relevant for finding diagnostic markers.

Figure 3.7 shows general good performance on the most important pathologies. Both hemorrhages and major parenchyma defects can be clearly delineated. However, the results also show points for improvement. In some cases, the hemorrhages on the derived NCCT do not have as good a contrast with the surrounding healthy brain tissue as on the reference NCCT, which could potentially hinder detection. Also, in some cases, the parenchyma defects near the border are less hypodense than on the reference NCCT and could be falsely identified as normal tissue. These effects are expected to be caused by having trained on predominantly normal cases. Also,

within the pathology cases, the majority of intracranial tissue can be considered normal as well. Therefore, it is expected that focused training on pathology samples with techniques like hard negative mining and increasing the number of pathology cases for training will help to further improve the robustness of the method and solve the currently observed issues.

Figure 3.5 shows the expected results between the baseline methods: t=0-3 performed best followed by t=0; t=all performed the worst. The earlier timepoints (t=0, t=0-3) showed less expression of the contrast agent – injection took place at approximately the first timepoint t=0 and required some time to circulate – and generally have a better signal-to-noise ratio, but a single timepoint (t=0) contains more noise than averaging over multiple timepoints (t=0-3) at the start of the 4D-CTA sequence.

The U-Net and C-LSTM models outperformed all the baseline methods, and the C-LSTM model outperformed all other trained models and baseline methods on all used evaluation metrics. Training and validating the C-LSTM on a bigger dataset showed similar performance, both for pathology and non-pathology cases Table 3.3. These results suggest a promising performance bound. However, pathology cases have only been validated as a whole and not on pathology-specific image quality.

The CNN baseline methods (the U-Net and the method of Nie et al.) use larger subvolumes for training than the C-LSTM method. However, it is not expected that a reduction in input subvolume size for training the CNN baseline methods will yield significantly different results. The choice for the bigger sub-volumes was based on the respective input sizes used for model training in their papers. The input sizes have been optimized for their respective methods. Using the same input size for all methods is expected to negatively influence the performance of the CNN baseline methods.

The parameter ablation results in Table 3.1 show that the best-performing parameterization (experiment 7) of the stacked C-LSTM model was a two-stack C-LSTM ($K = 2$) with an input kernel size of $*_x = 3^3$ and recurrent kernel size of $*_h = 3^3$. The best parameterization improved the performance of experiment 3, which had the same settings except that experiment 3 only used a single C-LSTM layer $K = 1$. Conversely, experiments 8 and 9 suggest that stacking more than two layers ($K > 2$) does not further improve the performance. We think that this effect is due to the exponential increase in optimization cost. Hence, it is likely that additional training might yield equivalent or better results. However, this was not feasible within this work. A hidden kernel size of $*_h = 3^3$ appeared to work better than one of 1^3. This was unexpected because the registered 4D-CTA data did not show much motion between timepoints, which would better justify the smaller recurrent kernel size. However, upon closer examination, the bigger kernel size might compensate for minor intra-

registration errors of the 4D-CTA timepoints with t> 0 to the first timepoint t=0. In general, larger kernel sizes are thought to work better since they allow for smoothing of the 4D-CTA image. Smoothing of the 4D-CTA will reduce the noise in the 4D-CTA. This is also the best what can be done to approximate the target NCCT from the 4D-CTA because the noise from the target NCCT is different from the 4D-CTA noise and cannot be estimated. Similarly, as with the $K > 2$ case, increasing the capacity beyond 3^3 to 5^5 yielded worse performance within the allotted number of training iterations due to the increased costs of optimization. In general, more stacks and bigger kernel sizes might be best. However, keeping a good balance between the feasibility of training optimization and regression performance, we found $K = 2$, an input kernel size of $*_x = 3^3$, and a recurrent kernel size of $*_h = 3^3$ to work best.

The C-LSTM is better suited for spatiotemporal data than CNN and LSTM methods separately. While it is possible to parse sequential data using CNNs[116], it is not a natural fit and requires some workarounds. Also, parsing spatiotemporal data with only LSTM using flattened spatial data ($*_x = 1^3$ and $*_h = x1^3$) would make it more difficult to encode translation-invariant spatial features (e.g., edges); this should result in poorer performance as was observed in experiment 1 from the hyperparameter ablation experiments in Table 3.1.

An interesting potential application of the C-LSTM model involves the optimization of the 4D-CTA acquisition protocol. As shown in Figure 3.8, the C-LSTM model results show that the final few timepoints t > 12 do not add much to the quantitative quality of the derived images and could be discarded to reduce patients' exposure to ionizing radiation. However, this work is limited to predicting the NCCT from 4D-CTA and the results might not translate to other prediction tasks from 4D-CTA like blood flow calculations. Nevertheless, the same method could be used for those applications as well under the condition that the model is first retrained for those tasks. Another limitation is the relatively small dataset it was trained and tested on. Note that, when starting with a pre-trained network, more optimization is required as the number of timepoints used deviates from the original training set.

The stroke workup for patients admitted to the hospital with suspicion of stroke could potentially be simplified wherein the majority of cases, a NCCT, a CTA, and often a 4D-CTA are taken. Our method and the method of Smit et al.[23] could be used to respectively derive a NCCT and a CTA from a 4D-CTA. Removing the need for a NCCT and a CTA, could potentially reduce radiation dose, workup time, and contrast usage. Yet, it is important to carefully investigate all of these factors and other factors like workup costs to assess the relevance for clinical routine adoption. However, this is beyond the scope of this work.

Regarding the minority situation where patients with suspicion of stroke will have

a hemorrhage (10% of all stroke patients) and patients are not eligible for alteplase admission, the workup would involve additional dose, workup time, and contrast usage over a conventional single NCCT, which is indeed a disadvantage compared to the conventional situation. However, if a 4D-CTA can now be used to derive a NCCT, we can still omit the administration of alteplase based on any hemorrhages identified on the derived NCCT. Furthermore, given this situation, it may be an added benefit for patients with both hemorrhagic and ischemic strokes, since the taken 4D-CTA images can be used to identify potential occlusions and estimate hemodynamics without additional costs for which a single NCCT is inadequate.

The proposed C-LSTM models and training scheme are not limited to deriving NCCT from 4D-CTA data and could be utilized for other applications involving spatiotemporal data. This work employed a regression scheme for training, but it is straightforward to make it into a segmentation scheme by adding a softmax to the model and changing the loss. In a preliminary study, we showed the feasibility of the C-LSTM model for whole volume training and prediction while maintaining context and expressiveness using gradient checkpointing[117]. The qualitative results from Figure 3.6 show that the C-LSTM model was able to suppress the vessels within the 4D-CTA, but it might also be used to filter other information as well. Another potential future direction is the use of the model for computing perfusion images.

A drawback of the proposed method is the computation time and memory requirements for training and prediction. It can take one or two weeks to train the best parameterization of the C-LSTM model from scratch; the evaluation can take up to forty minutes, which is not practical in a clinical setting. However, the model could be easily parallelized to divide the computational overhead during prediction. Furthermore, there is also speedup to be gained by implementing the model in more modern deep learning frameworks, e.g., using half-precision for the GPU computations. Training could be performed on larger inputs by using gradient checkpointing techniques[118] on each step function during the LSTM sequence. In conclusion, the computation time and memory requirements imposed great resource constraints, which limited the number of experiments. However, with modern solutions and better GPUs on the horizon, these issues will become less of a concern.

To further the acceptance of the method as a replacement for a normal NCCT scan, the evaluation of the derived images could be extended with a qualitative assessment of diagnostic relevant information (e.g., hemorrhages, dense vessel signs, and infarcts). This information could be graded for both the NCCT and the derived NCCT by experienced radiologists to assess whether all diagnostically relevant information is still present in the derived NCCT. However, this would require a larger dataset with more manually labeled pathology cases which is beyond the scope of

this work.

To conclude, we have presented the first deep learning application of C-LSTM for deriving 3D NCCT from 4D spatiotemporal CTA, which could potentially improve the efficiency of stroke workup. The proposed C-LSTM models and training scheme pose promising tools for handling spatiotemporal data in medical imaging and can be used for other problems as well.

4

MemCNN: a framework for memory efficient invertible networks

S.C. van de Leemput, J. Teuwen, B. van Ginneken, R. Manniesing

Original title: MemCNN: a Framework for Developing Memory Efficient Invertible Neural Networks

Published in: The Journal of Open Source Software, 4(39):1576, 2019

Abstract

Reversible operations have recently been successfully applied to classification problems to reduce memory requirements during neural network training. This feature is accomplished by removing the need to store the input activation for computing the gradients at the backward pass and instead reconstructing them on demand. However, current approaches rely on custom implementations of backpropagation, which limits applicability and extendibility. We present MemCNN, a novel PyTorch framework that simplifies the application of reversible functions by removing the need for customized backpropagation. The framework contains a set of practical generalized tools, which can wrap common operations such as convolutions and batch normalization and which take care of memory management. We validate the presented framework by reproducing state-of-the-art experiments using MemCNN and by comparing classification accuracy and training time on Cifar-10 and Cifar-100. Our MemCNN implementations achieved similar classification accuracy and faster training times while retaining compatibility with the default backpropagation facilities of PyTorch.

4.1 Introduction

Reversible functions, which allow exact retrieval of its input from its output, can reduce memory overhead when used within the context of training neural networks using backpropagation. That is since only the output is required to be stored, intermediate feature maps can be freed on the forward pass and recomputed from the output on the backward pass when required. Recently, reversible functions have been used with some success to extend the well-established residual network (ResNet) for image classification from He et al.[119] to more memory-efficient invertible convolutional neural networks[120–122] showing competing performance on datasets like Cifar-10, Cifar-100[123] and ImageNet[9]. However, the practical applicability and extendibility of reversible functions for the reduction of memory overhead have been limited, since current implementations require customized backpropagation, which does not work conveniently with modern deep-learning frameworks and requires substantial manual design.

The reversible residual network (RevNet) of Gomez et al.[120] is a variant on ResNet, which hooks into its sequential structure of residual blocks and replaces them with reversible blocks, that creates an explicit inverse for the residual blocks based on the equations from Dinh et al.[124] on nonlinear independent components estimation. The reversible block takes arbitrary nonlinear functions $\mathcal{F}$ and $\mathcal{G}$ and renders them invertible. Their experiments show that RevNet scores similar classification performance on Cifar-10, Cifar-100, and ImageNet, with less memory overhead.

Reversible architectures like RevNet have subsequently been studied in the framework of ordinary differential equations (ODE)[121]. Three reversible neural networks based on Hamiltonian systems are proposed, which are similar to the RevNet, but have a specific choice for the nonlinear functions $\mathcal{F}$ and $\mathcal{G}$ which are shown stable during training within the ODE framework on Cifar-10 and Cifar-100.

The i-RevNet architecture extends the RevNet architecture by also making the downscale operations invertible Jacobsen et al.[122], effectively creating a fully invertible architecture up until the last layer, while still showing good classification accuracy compared to ResNet on ImageNet. One particularly interesting finding shows that bottlenecks are not a necessary condition for training neural networks, which shows that the study of invertible networks can lead to a better understanding of neural network training in general.

The different reversible architectures proposed in the literature[120–122] have all been modifications of the ResNet architecture and all have been implemented in TensorFlow[125]. However, these implementations rely on custom backpropagation, which limits creating novel invertible networks and application of the concepts beyond the

application architecture. Our proposed framework MemCNN overcomes this issue by being compatible with the default backpropagation facilities of PyTorch. Furthermore, PyTorch offers convenient features over other deep learning frameworks like a dynamic computation graph and simple inspection of gradients during backpropagation, which facilitates inspection of invertible operations in neural networks.

In this work, we present MemCNN[1] a novel PyTorch[126] implementation which simplifies the use of reversible functions by removing the need for a customized backpropagation. MemCNN provides tools to drop-in memory-saving reversible functions within conventional PyTorch neural networks. Furthermore, it provides wrappers to convert arbitrary nonlinear functions to memory-saving reversible functions. We have validated the presented framework by implementing two state-of-the-art architectures (ResNet and RevNet) utilizing MemCNN, which are included in the GitHub repository, and compare them to existing state-of-the-art implementations in TensorFlow[125] on the Cifar-10 and Cifar-100 classification tasks on accuracy and training time. Our framework was found to achieve similar classification accuracy and faster training times. Validation experiments described in this work are included in the framework as well.

4.2 Methods

4.2.1 The reversible block

The core operator of MemCNN is the reversible block which is an operator that takes a function f and outputs a function $R : X \rightarrow Y$, and an inverse function $R^{-1} : Y \rightarrow X$ which resembles an invertible version of f. Here, $x \in X$ and $y \in Y$ can be arbitrary tensors with the same size and number of dimensions, i.e., $\mathrm{shape}(x) = \mathrm{shape}(y)$. Additionally, it must be possible to partition the input $x = (x_1, x_2)$ and output tensors $y = (y_1, y_2)$ in half, where each partition has the same shape, i.e., $\mathrm{shape}(x_1) = \mathrm{shape}(x_2) = \mathrm{shape}(y_1) = \mathrm{shape}(y_2)$. Formally, the reversible block operation (4.1), its inverse (4.2), and its partition constraints (4.3) provide a sufficiently general framework for implementing reversible operations.

For example, if one wants to create a reversible block performing a convolution followed by a ReLu f, the input $x \in X$ is partitioned in (x_1, x_2) of equal sizes to which this convolution block f is applied twice (say $\mathcal{F}$ and $\mathcal{G}$). The Reversible Block takes these two operators ($\mathcal{F}$ and $\mathcal{G}$) and outputs a "resblock"-like version R of the operator and an explicit inverse R^{-1}. Effectively the learnable function f is replaced by a learnable approximation R with an explicit inverse R^{-1}.

[1] MemCNN is available at: `https://github.com/silvandeleemput/memcnn`

$$R(x) = y \tag{4.1}$$
$$R^{-1}(y) = x, \tag{4.2}$$

with

$$\text{shape}(x_1) = \text{shape}(x_2) = \text{shape}(y_1) = \text{shape}(y_2) \tag{4.3}$$

4.2.2 Couplings

Using the above definitions we provide two different implementations for the reversible block in MemCNN, which we will call 'couplings'. A coupling provides a reversible mapping from (x_1, x_2) to (y_1, y_2). MemCNN supports two couplings: the additive coupling and the affine coupling.

Additive coupling

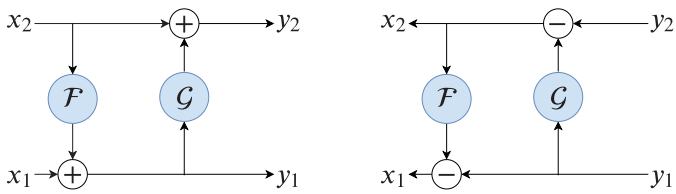


Figure 4.1: Graphical representation of additive coupling. The left graph shows the forward computations and the right graph shows its inverse. First, input x_1 and $\mathcal{F}(x_2)$ are added to form y_1, next x_2 and $\mathcal{G}(y_1)$ are added to form y_2. Going backwards, first, $\mathcal{G}(y_1)$ is subtracted from y_2 to obtain x_2; subsequently, $\mathcal{F}(x_2)$ is subtracted from y_1 to obtain x_1. Here, $+$ and $-$ stand for respectively element-wise summation and element-wise subtraction.

Equation 4.4 represents the additive coupling, which follows the equations of Dinh et al.[124] and Gomez et al.[120]. These support a reversible implementation through arbitrary (nonlinear) functions $\mathcal{F}$ and $\mathcal{G}$. These functions can be convolutions, ReLus, etc., as long as they have matching input and output shapes. The additive coupling is obtained by first computing y_1 from input partitions x_1, x_2 and function $\mathcal{F}$ and subsequently y_2 is computed from partitions y_1, x_2 and function $\mathcal{G}$. Next, (4.4) can be rewritten to obtain an exact inverse function as shown in (4.5). Figure 4.1 shows a graphical representation of the additive coupling and its inverse.

$$\begin{aligned} y_1 &= x_1 + \mathcal{F}(x_2), \\ y_2 &= x_2 + \mathcal{G}(y_1) \end{aligned} \tag{4.4}$$

$$\begin{aligned} x_2 &= y_2 - \mathcal{G}(y_1), \\ x_1 &= y_1 - \mathcal{F}(x_2) \end{aligned} \tag{4.5}$$

Affine coupling

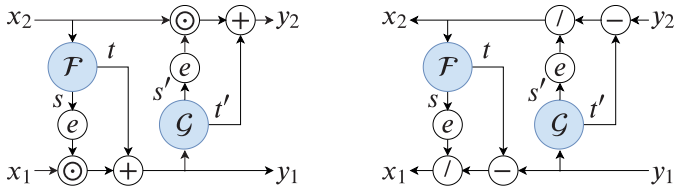


Figure 4.2: Graphical representation of the affine coupling. The left graph shows the forward computations and the right graph shows its inverse. Here, $\odot, /, +, -$, and e stand for element-wise multiplication, element-wise division, element-wise addition, element-wise subtraction, and element-wise exponentiation with base e respectively. First, s, t are computed for $\mathcal{F}(x_2)$, next input x_1 is element-wise multiplied with e^s and added to t to form y_1, subsequently s', t' are computed for $\mathcal{G}(y_1)$ and then x_2 is element-wise multiplied with $e^{s'}$ and added to t' to form y_2.

Equation 4.6 gives the affine coupling, introduced by Dinh et al.[127] and later used by Kingma and Dhariwal[128], which is more expressive than the additive coupling. The affine coupling, similar to the additive coupling, supports reversible implementations through arbitrary (nonlinear) functions $\mathcal{F}$ and $\mathcal{G}$. It also first computes y_1 from input partitions x_1, x_2 and function $\mathcal{F}$ and subsequently it computes y_2 from partitions y_1, x_2 and function $\mathcal{G}$. The difference with the additive coupling is that now the functions $\mathcal{F} = (s, t)$ and $\mathcal{G} = (s', t')$ each produce two equally sized partitions for scaling and translation, so $\text{shape}(x_1) = \text{shape}(s) = \text{shape}(t) = \text{shape}(s') = \text{shape}(t')$ holds. These components are then used to compute the output using element-wise product ($\odot$) and element-wise exponentiation with base e and element-wise addition ($+$). Equation 4.6 can be rewritten to obtain an exact inverse function as shown in equation 4.7, which uses element-wise division ($/$) and element-wise subtraction ($-$). Figure 4.2 shows a graphical representation of the affine coupling and its inverse.

$$y_1 = x_1 \odot e^s + t \text{ with } \mathcal{F}(x_2) = (s, t)$$
$$y_2 = x_2 \odot e^{s'} + t' \text{ with } \mathcal{G}(y_1) = (s', t') \tag{4.6}$$

$$x_2 = (y_2 - t')/e^{s'} \text{ with } \mathcal{G}(y_1) = (s', t')$$
$$x_1 = (y_1 - t)/e^s \text{ with } \mathcal{F}(x_2) = (s, t) \tag{4.7}$$

Implementation details

The reversible block has been implemented as a `torch.nn.Module` which wraps other PyTorch modules of arbitrary complexity for coupling functions $\mathcal{F}$ and $\mathcal{G}$. Each memory-saving coupling is implemented using at least one `torch.autograd.Function`, which provides a custom forward and backward pass that works with the automatic

differentiation system of PyTorch. Memory savings are implemented at the level of the reversible block and are achieved by setting the size of the underlying tensor storage to zero for inputs on the forward pass and restoring the storage size to the original size on the backward pass once it is required for computing gradients.

4.2.3 Building larger networks

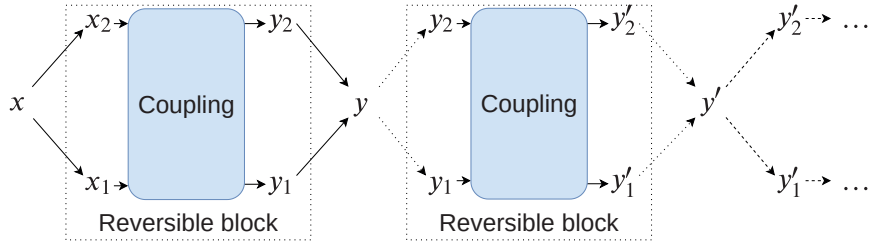


Figure 4.3: Graphical representation of chaining multiple reversible block layers.

The reversible block R can be chained by subsequent reversible blocks, e.g., $R_3 \circ R_2 \circ R_1$ for reversible blocks R_1, R_2, R_3, which creates a fully reversible chain of operations (see Figure 4.3). Additionally, reversible blocks can be mixed with regular functions f, e.g., $f \circ R$ or $R \circ f$ for reversible block R and regular function f. Note that mixing regular functions with reversible blocks often breaks the invertibility of reversible chains.

4.2.4 Memory savings

Technique	Authors	Memory Complexity	Computational Complexity
Naive		$O(L)$	$O(L)$
Checkpointing	[129]	$O(\sqrt{L})$	$O(L)$
Recursive	[118]	$O(\log L)$	$O(L \log L)$
Additive coupling	[120]	$O(1)$	$O(L)$
Affine coupling	[127]	$O(1)$	$O(L)$

Table 4.1: Comparison of memory and computational complexity for training a residual network (ResNet) between various memory saving techniques (extended table from Gomez et al. [120]). L depicts the number of residual layers in the ResNet.

The reversible block model has an advantageous memory footprint when chained in a sequence when training neural networks. After computing each $R(x) = y$ by (4.1)

on the forward pass, input x can be freed from memory and be recomputed on the backward pass, using the inverse function $R^{-1}(y) = x$ from (4.2). Once the input is restored, the gradients for the weights and the inputs can be recomputed as normal using the PyTorch 'autograd' solver. This effectively yields a memory complexity of $O(1)$ in the number of chained reversible blocks. Table 4.1 shows a comparison of memory versus computational complexity for different memory-saving techniques.

4.3 Experiments and results

Table 4.2: Accuracy (acc.) and training time (time, in hours:minutes) comparison of the PyTorch implementation (MemCNN) versus the Tensorflow implementation from Gomez et al.[120] on Cifar-10 and Cifar-100.[123]

	Tensorflow				PyTorch			
	Cifar-10		Cifar-100		Cifar-10		Cifar-100	
Model	**acc.**	**time**	**acc.**	**time**	**acc.**	**time**	**acc.**	**time**
resnet-32	92.74	2:04	69.10	1:58	92.86	1:51	69.81	1:51
resnet-110	93.99	4:11	73.30	6:44	93.55	2:51	72.40	2:39
resnet-164	94.57	11:05	76.79	10:59	94.80	4:59	76.47	3:45
revnet-38	93.14	2:17	71.17	2:20	92.80	2:09	69.90	2:16
revnet-110	94.02	6:59	74.00	7:03	94.10	3:42	73.30	3:50
revnet-164	94.56	13:09	76.39	13:12	94.90	7:21	76.90	7:17

To validate MemCNN, we reproduced the experiments from Gomez et al.[120] on Cifar-10 and Cifar-100[123] using their Tensorflow[125] implementation on GitHub[2], and made a direct comparison with our PyTorch implementation on accuracy and train time. We have tried to keep all the experimental settings, like data loading, loss function, train procedure, and training parameters, as similar as possible. All experiments were performed on a single NVIDIA GeForce GTX 1080 with 8GB of RAM. The results are listed in Table 4.2. Model performance of our PyTorch implementation obtained similar accuracy to the TensorFlow implementation with less training time on Cifar-10 and Cifar-100. All models and experiments are included in MemCNN and can be rerun for reproducibility.

Table 4.3 shows the average memory usage during model training using MemCNN. The results show significant memory savings using the invertible operations as

[2] `https://github.com/renmengye/revnet-public`

Table 4.3: GPU VRAM memory usage using the MemCNN implementation during training for all models. All models were trained on a NVIDIA GeForce GTX 1080 with 8GB of RAM. Significant memory savings were observed when using reversible operations as the number of layers increased.

Model	Layers	GPU VRAM
resnet	32	766 MB
resnet	110	1357 MB
resnet	164	3083 MB
revnet	38	677 MB
revnet	110	706 MB
revnet	164	1226 MB

the number of layers increases for ResNet without the use of invertible operations.

4.4 Works using MemCNN

MemCNN has recently been used to create reversible GANs for memory-efficient image-to-image translation by Ouderaa and Worrall[130]. Image-to-image translation considers the problem of mapping both $X \rightarrow Y$ and $Y \rightarrow X$ given two image domains X and Y using either paired or unpaired examples. In this work, the CycleGAN[109] model has been enlarged and extended with an invertible core using the reversible block, which they call RevGAN. Since the invertible core is weight-tied, training the model for the mapping $X \rightarrow Y$ automatically trains the model for mapping $Y \rightarrow X$. They show similar or increased performance of RevGAN with respect to similar non-invertible models like the CycleGAN with less memory overhead during training. The RevGAN model has also been applied to chest CT images[131].

4.5 Conclusion

We have presented MemCNN, a novel PyTorch framework, for creating and applying reversible operations for neural networks. It shows similar accuracy on Cifar-10 and Cifar-100 datasets with the current state-of-the-art method for reversible operations in Tensorflow and provides overall faster training times. The main features of the framework are smooth integration of reversible functions with other non-reversible functions by removing the need for a custom backpropagation and simple wrapping of arbitrary complex non-invertible nonlinear functions. The presented framework is

intended to facilitate the study and application of invertible functions in the context of neural networks.

5

Discussion

In this thesis, various deep-learning methods are presented to facilitate the analysis of 4D-CTA images in the context of stroke imaging. The first method focuses on the segmentation of the vasculature, white matter, gray matter, and cerebrospinal fluid in the cerebrum (**Chapter 2**). Secondly, a reconstruction method using a convolutional LSTM for a non-contrast CT (NCCT) from a 4D-CTA image is presented, which shows promising results (**Chapter 3**). Third, a PyTorch framework called MemCNN allows making arbitrary deep learning operations memory-efficient (**Chapter 4**). The presented methods and tools were initially developed to improve computer-aided diagnosis systems aimed at acute stroke analysis and to aid radiologists in reading 4D-CTA images. Not all of the work in this thesis is already directly usable in the clinical setting, but the results show promise and provide interesting insights and a basis for future research. The following sections elaborate on deep learning techniques and 4D-CTA scans for stroke imaging, discuss the presented methods, and present possible future research directions.

5.1 4D-CTA for acute stroke imaging

Within **Chapter 2** and **Chapter 3** the main focus was on 4D-CTA imaging for stroke imaging. However, this is currently not mainstream as it is not required by the standard stroke workup guidelines for thrombolytic treatment and thrombectomy decisions, and the current clinical acute stroke workup at most medical centers rely mostly on non-contrast CT (NCCT) and CT Angiography (CTA) imaging instead of 4D-CTA. However, it has been generally acknowledged that the additional diagnostic and prognostic information can help support clinical decision-making, mainly for the identification of potential patients eligible for endovascular treatment between 6-24 hours of symptom onset[132]. Also, while NCCT and CTA imaging also provide necessary information for stroke treatment, the benefit of 4D-CTA for stroke patient outcome and patient treatment selection has only recently been shown in various clinical trials[73–75]. Also, 4D-CTA requires high-end scanners, and not all hospitals have such equipment available in the emergency care department where stroke patients arrive. Likely, the stroke guidelines will further change in favor of more advanced imaging like 4D-CTA with insights gained from new research and large ongoing trials[133–135].

There is an increasing advocacy for using more 4D imaging within the stroke setting, because of the added dynamic information allowing for collateral flow estimation, which helps with the detection of vessel occlusions, core, and penumbra, and subsequent better treatment planning. Beyond these obvious use cases, 4D imaging also has great potential for the analysis of the vasculature and risk profiling[136], the analysis of brain tissue for better differentiation of the white- and gray matter[137]

(see also Chapter 2 of this thesis), help with studying and visualization of the blood flow dynamics[138], and individualized patient management based on extracted tissue values[139]. Furthermore, recent clinical evaluations of imaging for stroke-workup have argued for increased use of 4D-CTA for perfusion imaging after endovascular treatment procedures to predict long-term outcomes and identify opportunities for adjuvant therapy[140–142]. But there is also advocacy for using more advanced imaging (like 4D-CTA) within the early time window (within 4.5 hours of symptom onset) since detecting early ischemia signs on a NCCT scan is notoriously difficult[143].

CTA imaging has high sensitivity and specificity for large proximal vessel occlusions but has lesser reliability for more distal occlusions, which has led to the rise of multiphase CTA (mCTA)[144–146]. Multiphase CTA consists of a bolus-triggered single CTA head-neck acquisition, followed by two head CTA acquisitions. This type of imaging, just like the 4D-CTA, has multiple acquisitions over time, hence adding dynamic information for better estimating the blood flow. While the mCTA has less radiation dose, lower costs, and wider availability, the 4D-CTA has a better imaging resolution and can also be used to estimate capillary and venule level filling.

Currently, within the stroke setting there is no consensus on standardization for advanced imaging like 4D-CTA[143]. This is also one of the main reasons that, despite its potential, the 4D-CTA is currently not used that much in clinical practice, and general adaptation will require more automatization and standardization of imaging and stroke workup protocols[139]. For example, a practical limitation of the 4D-CTA data used for the works in this thesis typically only covered the brain and not the head-neck area as typically is the case for CTA. However, this practical issue could be addressed since research from our group has demonstrated the feasibility of acquiring a full head-neck 4D-CTA in the One-Step-Stroke protocol[147,148].

It can be argued that all the information of a CTA is also present within a properly acquired 4D-CTA scan. It has been shown for instance that the CTA can be derived from the 4D-CTA by a maximum intensity projection[23]. This observation is relevant in the context where both a CTA and a 4D-CTA scan are acquired; costs and patient exposure to ionizing radiation could be reduced by only administering a 4D-CTA scan. This could also speed up the workflow, which is important for stroke patients because ‘time is brain’. The individual timepoints within a 4D-CTA are typically acquired with less radiation than conventional stoke imaging like a NCCT or a CTA. However, the temporal information in a 4D-CTA image can be optimally estimated by computing a weighted temporal average (WTA) over all the timepoints weighted by their radiation dose to retrieve the optimal signal-to-noise where the total radiation dose encompasses the upper boundary of the image quality[30]. This essentially entails that it is possible to retrieve high-quality diagnostic images from 4D-CTA using less radiation

per timepoint than conventional CT techniques as long as the total radiation dose is sufficient.

5.2 Deep learning for acute stroke imaging

Medical image analysis has in the past decade rapidly shifted from traditional machine learning methods to deep learning[31]. Mostly due to the availability of larger datasets, parallelized computing power in the form of GPUs and other specialized hardware like TPUs, simplicity of application, and increased task performance. Within acute stroke imaging increasingly more data is available and the data gets increasingly larger and more complex. Meanwhile, the workload on radiologists is increasing worldwide and hence the time to analyze and process scans becomes less. Furthermore, with an increasingly aging population, the number of stroke patients is estimated to increase by 20-27% in the next 10-30 years in the European Union and the United States as well[149,150]. Hence, deep learning could potentially be utilized to speed up and alleviate tedious tasks of the radiologist, either serving as an extra set of eyes on the scan or potentially even operating as an expert reader that autonomously performs certain reading and or reporting tasks in the future.

For deep learning applications for stroke imaging various additional challenges arise. First of all, the processing speed of the stroke imaging applications is important, since timely diagnosis of stroke symptoms is crucial for successful patient outcome. Furthermore, the imaging data scale ranges from high-resolution NCCT and CTA scans to 4D dynamic CT scans, the domain has diverse and high-dimensional images for which small details might matter, which complicates fast processing and model training. That is, having large scans, with many voxels and dimensions generally leads to longer processing times and risks running into memory limits during model training with the available VRAM of the available GPUs. Research in this thesis has addressed these issues in several ways.

5.3 Beyond acute stroke imaging

While the main focus of this thesis has been researching and developing techniques for the acute stroke imaging setting, the work in this thesis is not necessarily limited to stroke imaging. 4D data can be found in other medical imaging domains and even beyond the medical imaging domain. Medical imaging will likely become increasingly dependent on 4D data in general. For example: 4D magnetic resonance imaging (MRI) and perfusion techniques for other organs. Hence techniques that are

developed for 4D stroke data likely translate to other organs and imaging modalities as well. Also, since the available data and the complexity of the data are increasing for many non-medical applications as well it is expected that techniques for dealing with high dimensional 4D data will become increasingly important for non-medical deep learning applications. In particular two points should be highlighted: ways to efficiently (quickly) extract relevant patterns from high dimensional data and ways to reduce the memory footprint during network training.

Chapter 2 and **Chapter 3** deal with how to extract relevant patterns directly from 4D data without performing any data reduction techniques beforehand. Working directly with 4D data generally makes the learning process more difficult and more time-consuming. However, an important benefit of this practice would be that the presented model can learn what data is relevant and/or could learn to perform dimensionality reduction on its own, preventing human-engineered biases. In the future, more research could focus on deep learning architectures for dealing with high-dimensional data as finding common deep learning techniques could be beneficial for many current and future applications.

Chapter 2 presents an optimized deep-learning method to segment white matter, gray matter, cerebrospinal fluid, and the vasculature for a whole brain 4D-CTA scan in 5 minutes on a GPU. Besides enabling the segmentation of essential functional brain areas, the model can learn relevant features directly from the 4D data instead of relying on manually derived 3D feature images, like a weighted temporal average (WTA) or weighted temporal variance (WTV). The presented model also shows better performance than the currently established state-of-the-art models. The method also should be able to produce better segmentations for the white matter and gray matter than can be achieved on a single CTA acquisition, since the additional dynamic information can be used to better delineate the vessels from the other functional structures. The model is also not limited to application to 4D-CTA scans and could be applied to other scans and imaging domains as well, although model retraining would be necessary.

In **Chapter 3** a deep learning method using 4D spatiotemporal convolutional LSTM is presented that allows reconstructing a 3D NCCT scan from a 4D-CTA scan. The model proves useful to investigate the utility and information contained within the 4D-CTA with respect to NCCT and CTA scans. While it seems likely that a CTA could potentially be replaced with a 4D-CTA acquisition, since it essentially contains almost the same information, it isn't commonly thought that an NCCT could potentially also be replaced by a 4D-CTA acquisition. However, the results of the model after having trained on several sets of coregistered NCCT / 4D-CTA pairs, show that this actually might be possible. If carefully evaluated, the stroke workup could po-

tentially be simplified by replacing the NCCT and CTA scans with a single 4D-CTA acquisition and deriving all the relevant information directly from the 4D-CTA using similar methods, reducing costs and reducing patient exposure to ionizing radiation. Yet, the big drawback is that this goes against standard practice.

In **Chapter 4** a way to reduce the memory footprint during neural network training was investigated. Memory requirements for neural network training, especially for high dimensional data like 4D, can be quite steep because the backward pass requires retaining the activations for each of its computations in memory. The consequence of this is that usually, some compromises have to be made by the deep learning practitioner regarding the network size, training speed, and target hardware during network design to deal with these constraints. In general memory footprint reduction techniques for deep learning network training (and inference as well) will be very helpful in making deep learning more practical and less of a burden on hardware and the environment.

For the works from **Chapter 2** and **Chapter 3** it should be noted that in both cases the maximum of 12 GB VRAM on a GPU, the ones available at the time of this research, used to train the models was quickly reached due to the large data size of the 4D-CTA data. Having practical memory limitations during model training can severely limit the possibilities of experimentation and/or limit the spatial or temporal contextual information given to a model. Thus, it became apparent that finding a way to reduce memory requirements during training would be important for deep learning practitioners. Reducing memory requirements will allow for larger models, processing larger inputs, and reducing the required technical workarounds and skills from the practitioner. Hence, in **Chapter 4** a deep learning framework implemented in PyTorch for allowing memory-efficient invertible operations was proposed. The PyTorch framework allows to wrap arbitrary invertible operations and renders them memory efficient, by discarding the activations during the forward pass of the training and recomputing them from the output during the backward pass using the inverse of the deep learning operation. Subsequently, largely based on the work from Gomez et al.[120], using so-called coupling operations, arbitrary non-invertible operations could be converted to invertible operations to render them memory efficient in the same way. We have made the software libraries created during this research freely available with a permissive open-source license, allowing others to use and build upon these ideas.

5.4 Future research

The work in this thesis provides a basis to further investigate and develop algorithms for the analysis of 4D-CTA. The methods from **Chapter 2** and **Chapter 3** can be used to analyze 4D data directly, but should be further validated on larger datasets and on more cases including pathologies and foreign objects. Especially the work from **Chapter 3** should be extended and investigated using an observer study to find out the quality of the reconstructed NCCT with respect to a conventional NCCT. The MemCNN framework from **Chapter 4** offers many directions for research into memory savings and invertible operations within deep learning and offers several opportunities for creating memory-efficient deep learning models.

The segmentation method from **Chapter 2** could be further improved by using additional training data and including tissue classes for pathology or foreign objects, such as core, penumbra, bleedings, clips, drains, calcifications, and bone if sufficiently annotated data for each class is collected. Also, scans from different scanners and acquisition protocols should be added to the training set. A remaining challenge here is the acquisition of a proper reference standard, which is validated by several radiologists to achieve a consensus among the tissue type labels. The segmentation model should be compared with other newer state-of-the-art segmentation frameworks like nnU-Net[151] to further validate the found architectural improvements and training method. Finally, since the model is not limited to the segmentation of structures in 4D-CTA only, it can be applied in other domains to test its segmentation performance against other methods.

The technical work from **Chapter 3** should be further tested in an observer study to verify the usability of the method for clinical practice, which could move the field towards one-step stroke image analysis. Experienced radiologists could grade the quality of a regular NCCT and the reconstructed NCCT scans to see if the latter could be used as a replacement by evaluating, for example, diagnostic relevant information and/or image quality. Early experiments that we carried out in our group showed a potential problem with the reconstructed areas around the cerebellum, which might be due to the reduced individual quality of each timepoint within the 4D-CTA, but is typically an important area within an NCCT to inspect. Another important factor is that a typical hemorrhage is an important bright-appearing phenomenon to look for on NCCT scans. However, the reconstructed NCCT partly focuses on the suppression of bright-appearing vasculature, hence proper validation and potential additional training should be performed to ensure proper behavior. Potentially, these limitations could be addressed by collecting a much larger and more diverse set of training data. New training data in the form of NCCT and 4D-CTA images can also be acquired us-

ing modern reconstruction kernels for CT, which in turn should result in better-quality reconstructions after model training.

The convolutional LSTM model used in **Chapter 3** provides an interesting direction for dealing with the spatiotemporal dynamic nature of 4D-CTA data. The convolutional component of the model is typically efficient in dealing with spatial features, whereas the recurrent part of the model is typically used to deal with temporal features. In our work, we fixed the number of timepoints to 19 as this was common protocol within our hospital, but the model is not limited to a fixed number of timepoints and could help determine the necessary number of acquisitions required for optimal signal retrieval from a 4D-CTA.

A lot of factors about C-LSTM models require more research. For example, how to best train and optimize them, how to design them best, and for which applications they are most suited. Furthermore, since the model is not as popular as conventional CNNs, a broad range of applications using the model are still unexplored. Subsequent research can focus on applying such models for 3D image analysis, processing 2D slices using the recurrent mechanism to process the slices sequentially. Here lies a great potential for increasing the receptive field of the model network along the axis chosen for recursion. Also, the model should have the capacity to capture long-term dependencies in the data better, which could be helpful for ordered segmentation labeling tasks. Processing 3D images in this manner could also help to reduce memory requirements during training[117], by for example checkpointing[118] various steps within the recurrent mechanism. Finally, in recent years, the transformer model[152] has become successful and popular for dealing with sequential data like next token prediction. It would be interesting to investigate if a similar model can be adapted and or extended to handle medical images with temporal or sequential components.

The freely available MemCNN PyTorch framework from **Chapter 4** provides a simple interface for creating memory-efficient operations for model network training and has shown a few successful applications[130,131]. The applied work with MemCNN highlights the applicability of invertible operations for generative adversarial deep learning networks for domain adaptation in particular, due to their built-in invertible nature. However, several factors limit the usability of the framework. First of all, making an operation invertible imposes restrictions on the input and output size of an operation: the input and output tensors have to possess the same number of elements. Secondly, the coupling operations alter the output signal significantly to ensure invertibility, which hinders the interpretability of the operations. Finally, within the domain of medical image analysis where segmentation models with many shortcut connections are popular (e.g., U-Net[51]) the applicability of invertible operations

is limited since the interconnected nature of those architectures limits the number of activations that can be rendered memory-efficient (i.e., it works best if all operations can be coupled in series). Hence, more research is needed to apply invertible operations for popular models like U-Net and to investigate if it is possible to make operations invertible that do not have the same number of elements in the input and output. In conclusion, more work is needed to further understand and interpret invertible operations in the context of deep learning.

Further advances in memory optimizing methods and the availability of better and more specialized hardware will allow training on bigger datasets and data. This is a nice prospect as the number of 4D-CTA scans acquired keeps growing, and with the addition of even higher resolution scanners, each 4D-CTA scan itself keeps growing in size. While hardware is improving and data size is growing, we are faced with the continuous challenge of collecting good-quality reference standards for all the available data. Here we can take advantage of what we have already built. Pre-trained models like the one from **Chapter 2** can be used to weakly label existing data, which can be manually corrected by (preferably multiple) expert readers. The newly labeled data can subsequently be used to retrain or tune the models. However, organizing and implementing such efforts in a multi-center setting remains a challenge.

5.5 Concluding remarks

In this thesis, I have presented various deep-learning methods for the analysis of 4D-CTA scans for acute stroke imaging. Furthermore, I provide a deep-learning PyTorch framework called MemCNN for deep-learning practitioners to reduce memory requirements during neural network training. The methods and tools presented in this thesis can serve as a basis for subsequent research and for the development of better methods for 4D-CTA in the context of stroke image analysis.

Summary
Samenvatting
Publications
Data management
PhD portfolio
Bibliography
Dankwoord
Curriculum Vitae

Summary

Acute stroke is the second leading cause of death and is the third leading cause of disability worldwide. Stroke is caused by a disturbance of blood flow in the brain which can result in a loss of brain function. Hemorrhagic stroke refers to the rupture of blood vessels in the brain, which comprises approximately 13% of all stroke cases[2]. Ischemic stroke is caused by the blockage of blood vessels by a thrombus, usually in the form of a blood clot, which accounts for approximately 87% of all stroke cases[2]. Survivors of stroke often suffer from various complications both neurological and physical and it is estimated that 6.5 million people will die as a result of stroke each year[1].

Computed tomography (CT) is the primary imaging modality for quick assessment of cerebral conditions due to its widespread availability, low cost, and high speed[4]. A non-contrast CT (NCCT) scan can provide a quick visualization of the brain areas and help to identify potential pathology like hemorrhagic strokes. Other techniques like CT angiography (CTA), which involves the injection of a radiocontrast agent briefly before making the scan, allow to visualize and assess the cerebral vasculature. Four-dimensional CTA (4D-CTA) encompasses multiple scans in rapid succession over time while still relying on the injection of a radiocontrast agent. This results in a dynamic sequence of 3D images, which allow visualization and assessment of cerebral blood flow and volume in patients suspected of acute stroke.

Patients suffering from a stroke require fast diagnosis and treatment to minimize brain damage and maximize patient outcomes. Hence, because of the quick acquisition speed, CT is the standard image modality for acute stroke imaging. The first diagnostic priority is to differentiate between hemorrhagic and ischemic stroke for which a non-contrast CT (NCCT) scan is performed. When a hemorrhagic stroke or lesions can be ruled out, CTA and/or 4D-CTA scans can be taken to identify blood-deprived areas in the brain and find potential causes and locations of thrombi. 4D-CTA acquisitions contain additional dynamic information over CTA, which makes them an interesting yet challenging source of information for stroke image analysis.

The amount of work analyzing images carried out by neuroradiologists is getting increasingly time-consuming and tedious due to an increasing number of patients, the increase of imaging data, and the increase of data with higher spatial and temporal resolutions. Hence, machine learning methods that can support radiologists, for example with computer-aided diagnosis (CAD) systems, and automate parts of the diagnosis are becoming increasingly relevant. In particular deep learning, which are a set of machine learning methods that allow learning task-related features directly from data without requiring algorithm developers to explicitly provide or extract

task-relevant features beforehand.

Deep learning models are data-driven algorithms that have to be optimized for a certain goal using a training process to perform well on one or more performance metrics. Deep learning methods are often characterized by requiring a lot of processing power and will often demand specialized hardware like graphics processing units (GPUs) and tensor processing units (TPUs) to perform model inference and especially model training in a timely manner. For processing large inputs, like the 4D-CTA acquisitions in this thesis, the current state-of-the-art training methods of deep learning models generally require allocating a lot of memory on the specialized hardware, which often limits the scope of the deep learning applications and forces practitioners to make concessions in their approach.

In this thesis, two different deep learning methods are presented for the analysis of 4D-CTA images. The first method is focused on brain tissue segmentation and the second is on NCCT reconstruction. Finally, a deep learning framework called MemCNN is presented which allows for memory-efficient training of deep learning networks.

Chapter 2 describes a method for 3D segmentation of white matter, gray matter, cerebrospinal fluid, and cerebral vasculature in 4C CT images. A modified U-Net deep learning architecture was trained and validated on 42 4D-CTA acquisitions of the brain of patients with suspicion of acute ischemic stroke, for which the data was annotated by two trained observers using 2D sparse annotations. The model performance was validated on dice coefficient, contour mean distance, and absolute volume difference. Finally, the performance of the model was estimated on a separate fully annotated test set of 5 cases by the same observers. The performance metrics were found to be similar to the average interobserver variability scores and that it was outperforming the current state-of-the-art. The results showed that the modifications made to the U-Net contributed to significantly better segmentation performance and that it is possible to learn a state-of-the-art model directly end-to-end from the 4D data without using intermediate 3D representations like the weighted temporal average (WTA) and weighted temporal variance (WTV).

Chapter 3 presents a method for performing a 3D non-contrast CT reconstruction from 4D-CTA data using a stacked bidirectional convolutional LSTM (C-LSTM) network. This method could potentially be used to simplify the imaging workup in acute stroke resulting in reduced workup time and radiation dose. Several parameterizations of the C-LSTM network were trained on a set of 17 4D-CTA/NCCT pairs to learn to derive an NCCT from a 4D-CTA and were subsequently quantitatively evaluated on a cohort of 16 pairs. The results show that the C-LSTM model clearly outperforms the baseline and other competitive convolutional neural network methods. The work

shows good scalability and performance of the method by continued training and testing on an independent dataset which includes pathology of 80 and 83 4D-CTA/NCCT pairs, respectively. The presented C-LSTM model is, therefore, a promising general deep-learning approach to learning from high-dimensional spatiotemporal medical images.

Chapter 4 describes MemCNN, a PyTorch framework that provides tools for invertible deep learning operations. Invertible deep learning operations have recently been successfully applied to classification problems to reduce memory requirements during neural network training. The core functionality is implemented as a customized backpropagation step that can be used for any invertible PyTorch function which takes care of memory management. In addition, MemCNN provides a set of couplings to convert non-invertible operations into invertible operations, such as convolutions and batch normalization. The presented framework was validated by reproducing state-of-the-art experiments by comparing classification accuracy and training time on Cifar-10 and Cifar-100. MemCNN implementations achieved similar classification accuracy and faster training times while retaining compatibility with the default backpropagation facilities of PyTorch.

In summary, the methods presented in this thesis form a basis for the further development of algorithms for the automatic analysis of 4D-CTA in stroke imaging. The cerebral vasculature and other relevant brain areas can be automatically segmented and labeled, an NCCT can be automatically derived from a 4D-CTA potentially simplifying the stroke workup, and more memory-efficient network training can be achieved using the presented open-source MemCNN PyTorch framework. Further research and developments in this field should lead to more systems and tools that can ultimately find their way into clinical environments, where they can support clinical physicians with disease diagnosis and treatment planning.

Samenvatting

De acute beroerte is wereldwijd de twee na grootste oorzaak van overlijden en is de drie na grootste oorzaak van invaliditeit[1]. Een beroerte wordt veroorzaakt door een verstoring van de bloedstroom in de hersenen die kan leiden tot verlies van hersenfunctie. Een hemorragische beroerte is een breuk van de bloedvaten in de hersenen en komt in ongeveer 13% van alle beroertes voor[2]. Een ischemische beroerte wordt veroorzaakt door een verstopping van de bloedvaten, meestal door een bloedprop, en is verantwoordelijk voor ongeveer 87% van alle beroertes[2]. Patiënten die een beroerte hebben gehad hebben vaak last van verschillende neurologische en fysieke complicaties. Er wordt geschat dat er elk jaar 6.5 miljoen mensen sterven aan een beroerte[1].

Computertomografie (CT) is de primaire beeldvormingsmodaliteit voor het snel beoordelen van hersenaandoeningen vanwege de brede inzetbaarheid, de lage kosten, en de hoge snelheid van scannen[4]. Een standaard CT-scan zonder contrast (NCCT) geeft een snelle visualizatie van hersengebieden en kan helpen bij het identificeren van potentiele aandoeningen zoals hemorragische beroertes. Andere technieken zoals CT-angiografie (CTA), waarbij een injectie met radiocontrastmiddel wordt gebruikt kort voordat de scan wordt gemaakt, maken het mogelijk om de cerebrale bloedvaten te visualizeren. Vierdimensionale CTA (4D-CTA) bestaat uit meerdere scans die kort na de injectie met radiocontrastmiddel genomen worden. Dit resulteert in een dynamische sequentie van 3D beelden, die kan helpen bij de visualizatie en kwantificatie van de cerebrale bloedstroom en eventuele perfusiedefecten bij patiënten met een verdenking op een acute beroerte.

Patiënten met een beroerte hebben een snelle diagnose en behandeling nodig om zo de hersenschade te beperken met daardoor het beste perspectief op herstel. Dankzij de snelle acquisitietijden is CT de standaard beeldvormingsmodaliteit voor het beoordelen van acute beroertes. De eeste diagnostische prioriteit is om het verschil tussen een hemorragische en een ischemische beroerte vast te stellen, waarvoor een CT-scan zonder contrast (NCCT) wordt gemaakt. Als hemorragische beroerte of andere pathalogische aandoeningen kunnen worden uitgesloten, worden CTA- en/of 4D-CTA-scans afgenomen om te zien welke hersengebieden een tekort aan bloedtoevoer hebben en zo potentiële oorzaken zoals bloedproppen te lokaliseren. In vergelijking met CTA hebben 4D-CTA scans het bijkomende voordeel dat ze dynamische informatie vastleggen. Dit maakt de 4D-CTA scan een interessante maar uitdagende bron van beeldinformatie voor de analyse van beroertes.

De hoeveelheid beelden die de neuroradiologen moeten analyseren neemt toe door het stijgende aantal patiënten, de toename van beelddata, en de toename van

beelddata met hogere spatiële en temporele resoluties. Het is daarom van belang om meer werk uit handen van de radioloog te nemen en om het werk te versimpelen door geautomatiseerde ondersteuning te bieden met behulp van computer aided diagnosis (CAD) systemen en machine learning toepassingen. Deep learning is een vorm van machine learning die in staat is om taakgerelateerde kenmerken direct uit de data te leren zonder dat een programmeur deze expliciet meegeeft aan een algoritme.

Deep learning algoritmes zijn datagebaseerde algoritmes die moeten worden geoptimaliseerd voor een bepaald doel door middel van een trainprocedure, zodat ze goed scoren op één of meer prestatiemetingen. Deep learning algoritmes vereisen vaak aanzienlijke computerkracht, waardoor deze algoritmes meestal op gespecialiseerde hardware zoals graphic processing units (GPUs) en tensor processing units (TPUs) worden uitgevoerd zodat inferentie en vooral de trainprocedures enigzins snel verlopen. De state-of-the-art deep learning algoritmes vragen ook veel geheugen voor het verwerken van grote invoer, zoals de 4D-CTA beelden in dit proefschrift. Dit beperkt de omvang van de mogelijke deep learning applicaties en dwingt de algoritmemakers tot concessies.

In dit proefschrift worden twee verschillende deep learning methodes beschreven voor de analyse van 4D-CTA-beelden. De eerste methode is gericht op de segmentatie van beroertegerelateerde hersengebieden en de tweede methode is gericht op het reconstrueren van een NCCT-beeld uit een 4D-CTA-hersenscan. Daarnaast presenteren we in dit proefschrift een open-source deep learning PyTorch framework genaamd MemCNN, waarmee geheugenefficiënte deep learning netwerken kunnen worden gemaakt en getraind.

Hoofdstuk 2 beschrijft een deep learning methode voor het segmenteren van de witte materie, de grijze materie, de cerebrospinale vloeistof, en het hersenvatenstelsel op 4D-CTA-beelden. Hiervoor werd een gemodificeerd 3D U-Net deep learning netwerk getraind en gevalideerd op 42 4D-CTA-hersenscans van patiënten met een verdenking op een acute ischemische beroerte. De data werden geannoteerd door twee getrainde beoordelaars (observers) door gebruik te maken van één enkele 2D plak per scan. Het model werd geëvalueerd met behulp van dice coefficient, contour mean distance, en absolute volume difference als uitkomstmaten en werd bovendien geëvalueerd op een aparte volledig geannoteerde testset van 5 hersenscans. De uitkomstmaten bleken vergelijkbaar met de gemiddelde verschillen tussen de observers en waren beter dan de huidige state-of-the-art. De resultaten lieten verder zien dat de aanpassingen aan het 3D U-Net tot significant betere segmentaties leiden en dat het mogelijk is om het model direct uit 4D data te leren zonder afgeleide 3D beelden zoals de weighted temporal average (WTA) en de weighted temporal

variance (WTV) te gebruiken.

Hoofdstuk 3 toont een methode om een 3D CT hersenscan zonder contrast te reconstrueren uit een 4D-CTA-hersenscan door gebruik te maken van meerdere gestapelde bidirectional convolutional LSTM (C-LSTM) deep learning netwerken. De gepresenteerde methode kan in potentie gebruikt worden om de huidige richtlijnen voor beeldanalyse op het gebied van beroerte bij te stellen en zo een versimpeling, reductie van doorlooptijd, en reductie van toegediende straling te realiseren. Verschillende C-LSTM netwerken werden getraind op 17 4D-CTA/NCCT-paren en werden vervolgens kwantitatief en kwalitatief geëvalueerd op een aparte cohort van 16 paren. De resultaten lieten zien dat het C-LSTM-model betere resultaten behaalt dan competitieve deep learning methodes. Verder lieten we zien dat de methode ook kan worden opgeschaald naar grotere datasets met pathologie, door de getrainde methode verder te trainen op een aparte onafhankelijke dataset met 80 4D-CTA/NCCT-paren en door dit te evaluaten op 83 paren. De gepresenteerde deep-learning methode lijkt dan ook veelbelovend om toe te passen op hoogdimensionale spatiotemporele medische afbeeldingen.

Hoofdstuk 4 beschrijft MemCNN: een open-source PyTorch framework voor het maken en trainen van inverteerbare deep learning netwerken. Inverteerbare deep learning netwerken zijn recentelijk met succes toegepast op classificatieproblemen om gedurende het trainen van neurale netwerken geheugenvereisten te verminderen. De kernfunctionaliteit van MemCNN is een aangepaste geheugenefficiënte backpropagationstap die werkt op willekeurige inverteerbare PyTorch functies die het geheugengebruik reguleert. Daarnaast biedt MemCNN een tweetal koppelingen (in de vorm van PyTorch functies) om niet-inverteerbare operaties om te zetten in inverteerbare operaties, zoals convoluties en batch normalisatie. Het raamwerk is gevalideerd door een aantal state-of-the-art experimenten te reproduceren op de Cifar-10 en Cifar-100 datasets en door de classificatienauwkeurigheid en de benodigde trainingstijd te beoordelen. De implementaties die gebruik maakten van MemCNN behaalden een vergelijkbare classificatienauwkeurigheid en snellere trainingstijden. Doordat MemCNN gebruikt kan worden met de standaard backpropagation van PyTorch, is het makkelijk te gebruiken en toe te passen voor bestaande neurale netwerken.

De gepresenteerde methodes in dit proefschrift vormen een basis voor de verdere ontwikkeling van algoritmes voor de automatische analyse van 4D-CTA-hersenscans voor beroertes. Het hersenvatenstelsel en andere relevante hersenstructuren kunnen automatisch worden gesegmenteerd en gelabeled, een NCCT kan automatisch worden berekend van een 4D-CTA-hersenscan wat mogelijk kan leiden tot vereenvoudiging van de beroertebeeldvormingsrichtlijnen, en deep learning netwerken kun-

nen op een geheugenefficiënte wijze worden getraind met behulp van de open-source MemCNN PyTorch framework. Verder onderzoek en ontwikkelingen op dit gebied zullen leiden tot het maken van systemen die uiteindelijk hun weg zullen vinden in de klinische werkomgeving waar ze de artsen kunnen helpen met het stellen van nauwkeurige diagnoses en het inschatten van prognoses.

Publications

Papers in international journals

M. Meijs, S.A.H. Pegge, A. Patel, **S.C. van de Leemput**, K. Koschmieder, L. Vos, F.J.A. Meijer, M. Prokop, R. Manniesing "Cerebral Artery and Vein Segmentation in Fourdimensional CT Angiography Using Convolutional Neural Networks". *Radiology: Artificial Intelligence*, 2(4):e190178, 2020.

S.C. van de Leemput, M. Prokop, B. van Ginneken, R. Manniesing. "Stacked Bidirectional Convolutional LSTMs for Deriving 3D Non-contrast CT from Spatiotemporal 4D CT". *IEEE Transactions on Medical Imaging*, 39(4):985-996, 2019.

S.C. van de Leemput, J. Teuwen, B. van Ginneken, R. Manniesing. "MemCNN: A Python/PyTorch Package for Creating Memory-efficient Invertible Neural Networks". *Journal of Open Source Software*, 4(39):1576, 2019.

S.C. van de Leemput, M. Meijs, A. Patel, F.J.A. Meijer, B. van Ginneken, R. Manniesing. "Multiclass Brain Tissue Segmentation in 4D CT using Convolutional Neural Networks". *IEEE Access*, 7:51557-51569, 2019.

A. Patel, **S.C. van de Leemput**, M. Prokop, B. van Ginneken, R. Manniesing. "Image Level Training and Prediction: Intracranial Hemorrhage Identification in 3D Non-Contrast CT". *IEEE Access*, 7:92355-92364, 2019.

M. Meijs, A. Patel, **S.C. van de Leemput**, M. Prokop, E.J. van Dijk, F.E. de Leeuw, F.J.A. Meijer, B. van Ginneken, R. Manniesing. "Robust Segmentation of the Full Cerebral Vasculature in 4D CT Images of Suspected Stroke Patients". *Nature: Scientific Reports*, 7:15622, 2017.

Papers in conference proceedings

S.C. van de Leemput, A. Patel, R. Manniesing. "Full Volumetric Brain Tissue Segmentation in Non-contrast CT using Memory Efficient Convolutional LSTMs". In: *Conference on Neural Information Processing Systems, Medical Imaging Meets NeurIPS workshop track*, 2018.

S.C. van de Leemput, M. Meijs, A. Patel, B. van Ginneken, M. Prokop, R. Manniesing. "Stacked Bidirectional Convolutional LSTMs for 3D Non-contrast CT Reconstruction from Spatiotemporal 4D CT". In: *International Conference on Medical*

Imaging with Deep Learning, 2018.

S.C. van de Leemput, J. Teeuwen, R. Manniesing. "MemCNN: a Framework for Developing Memory Efficient Deep Invertible Networks". In: *International Conference on Learning Representations, Workshop Track*, 2018.

A. Patel, **S.C. van de Leemput**, M. Prokop, B. van Ginneken, R. Manniesing. "Automatic Cerebrospinal Fluid Segmentation in Non-Contrast CT Images Using a 3D Convolutional Network". In: *Medical Imaging*, volume 10134 of Proceedings of the SPIE, 2017.

S.C. van de Leemput, F. Dorssers and B.E. Bejnordi. "A novel spherical shell filter for reducing false positives in automatic detection of pulmonary nodules in thoracic CT scans". In: *Medical Imaging*, volume 9414 of Proceedings of the SPIE, 2015.

Abstracts in conference proceedings

M. Meijs, A. Patel, **S.C. van de Leemput**, B. van Ginneken, M. Prokop, R. Manniesing. "Vessel Segmentation using Deep Learning". In: *Annual Meeting of the Radiological Society of North America*, 2018.

S.C. van de Leemput, F. J. A. Meijer, M. Prokop, R. Manniesing. "Cerebral white matter, gray matter and cerebrospinal fluid segmentation in CT using VCAST: a volumetric cluster annotation and segmentation tool". In: *European Congress of Radiology*, 2017.

R. Manniesing, **S.C. van de Leemput**, M. Prokop and B. van Ginneken. "White Matter and Gray Matter Segmentation in 4D CT Images of Acute Ischemic Stroke Patients: a Feasibility Study". In: *Annual Meeting of the Radiological Society of North America*, 2016.

Data management

All the primary and secondary data obtained during my PhD at the Radboud university medical center (Radboudumc) have been captured as anonymized data archives which are centrally stored and backed up daily on the local Radboudumc server. All data archives are accessible by the associated senior staff members. For each published article, the source code, package dependencies, and additional files such as method parameters are compiled and stored in specialized containers that can be run on local and cloud-based hardware. This ensures that the published method can be used to reproduce all results or be used on previously unseen data. All project source code and documentation, including an in-depth description of the experiments performed, is backed up using a secure cloud-based service with version control. Our research group provides a platform for grand challenges in medical image analysis which allows other researchers to easily compare the performance of their algorithms using an automated evaluation of a dataset corresponding to a publication. All aforementioned practices adhere to the FAIR data principles.

PhD portfolio

Name PhD candidate:	S.C. van de Leemput
Department:	Radiology and Nuclear Medicine
Graduate School:	Radboud Institute for Health Sciences
PhD period:	01-08-2015 - 01-08-2019
Promotors:	Prof. dr. M. Prokop
	Prof. dr. ir. B. van Ginneken
Copromotor:	Dr. ir. R. Manniesing

Institute for Health Sciences
Radboudumc

	Year(s)	ECTS
Training activities		
a) Courses & Workshops		
- Scientific Integrity	2018	1.0
- Scientific Writing for PhD Candidates	2016	3.0
- Deep Learning 101 Workshop	2016	3.0
- RIHS introduction course for PhD students	2016	1.0
- NFBIA Front End Vision	2015	6.0
- NFBIA Summer School	2015	1.5
- General introduction day Radboudumc	2015	0.3
b) Seminars & lectures		
- NFBIA symposium	2017	0.3
- Annual DIAG-FME symposium	2016	1.0
c) Symposia & congresses		
- NeurIPS Conference on Neural Information Processing Systems†	2018	3.0
- MIDL International Conference on Medical Imaging with Deep Learning†	2018	2.0
- ICLR International Conference on Learning Representations‡	2018	3.0
- ECR European Congress of Radiology †	2017	3.0
d) Other		
- Weekly research meeting	2015-2019	7.0
- Weekly DIAG discussion hour	2015-2019	7.0
Total		**42.1**

†Indicates a poster and an oral presentation
‡Indicates a poster presentation

Radboud University
Radboudumc university medical center

Bibliography

[1] Feigin V. L., Brainin M., Norrving B., Martins S., Sacco R. L., Hacke W., Fisher M., Pandian J., and Lindsay P. World Stroke Organization (WSO): Global Stroke Fact Sheet 2022. *Int J Stroke*, 17(1):18–29, Jan 2022.

[2] American stroke association - types of stroke. `https://www.stroke.org/en/about-stroke/types-of-stroke`, 2023. Accessed: 2023-09-11.

[3] Saver J. L. Time is brain — quantified. *Stroke*, 37(1):263–266, 2006.

[4] Bellolio M. F., Heien H. C., Sangaralingham L. R., Jeffery M. M., Campbell R. L., Cabrera D., Shah N. D., and Hess E. P. Increased computed tomography utilization in the emergency department and its association with hospital admission. *Western Journal of Emergency Medicine*, 18(5):835, 2017.

[5] Röntgen W. C. On a new kind of rays. *Science*, 3(59):227–231, 1896.

[6] Mitchell T. M. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[7] Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[8] LeCun Y., Bottou L., Bengio Y., and Haffner P. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[9] Deng J., Dong W., Socher R., Li L.-J., Li K., and Fei-Fei L. ImageNet: A large-scale hierarchical image database. In *CVPR09*, 2009.

[10] Rumelhart D. E., Hinton G. E., and Williams R. J. *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988. ISBN 0262010976.

[11] Cauchy A. et al. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[12] Robbins H. and Monro S. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[13] Kiefer J. and Wolfowitz J. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952.

[14] Meijs M., Patel A., van de Leemput S., Prokop M., van Dijk E. J., de Leeuw F.-E., Meijer F. J. A., van Ginneken B., and Manniesing R. Robust segmentation of the full cerebral vasculature in 4D CT images of suspected stroke patients. *Nature Scientific Reports*, 7, 2017.

[15] Meijs M., de Leeuw F.-E., Boogaarts H. D., Manniesing R., and Meijer F. J. A. Circle of willis collateral flow in carotid artery occlusion is depicted by 4D-CTA. *World Neurosurgery*, 114: 421–426, 2018.

[16] Meijs M., Pegge S. A., Murayama K., Boogaarts H. D., Prokop M., Willems P. W., Manniesing R., and Meijer F. J. Color mapping of 4D-CTA for the detection of cranial arteriovenous shunts. *American Journal of Neuroradiology*, 40(9):1498–1504, 2019.

[17] Meijs M., Meijer F. J. A., Prokop M., van Ginneken B., and Manniesing R. Image-level detection of arterial occlusions in 4D-CTA of acute stroke patients using deep learning. *Medical Image Analysis*, 66:101810, 2020.

[18] Patel A., van Ginneken B., Meijer F. J. A., van Dijk E. J., Prokop M., and Manniesing R. Robust cranial cavity segmentation in CT and CT perfusion images of trauma and suspected stroke patients. *Med. Image Anal.*, 36:216–228, 2016.

[19] Patel A., van Ginneken B., Meijer F. J. A., van Dijk E. J., Prokop M., and Manniesing R. Robust cranial cavity segmentation in ct and ct perfusion images of trauma and suspected stroke patients. *Medical Image Analysis*, 36:216–228, feb 2017.

[20] Patel A., van de Leemput S. C., Prokop M., van Ginneken B., and Manniesing R. Image level training and prediction: Intracranial hemorrhage identification in 3d non-contrast ct. *IEEE Access*, 7:92355–92364, 2019.

[21] Patel A., Schreuder F. H. B. M., Klijn C. J. M., Prokop M., van Ginneken B., Marquering H. A., Roos Y. B. W. E. M., Baharoglu M. I., Meijer F. J. A., and Manniesing R. Intracerebral haemorrhage segmentation in non-contrast ct. *Nature Scientific Reports*, 9:17858, 11 2019.

[22] Larson D. B., Johnson L. W., Schnell B. M., Salisbury S. R., and Forman H. P. National trends in CT use in the emergency department: 1995–2007. *Radiology*, 258(1):164–173, 2011.

[23] Smit E. J., Vonken E.-J., van Seeters T., Dankbaar J. W., van der Schaaf I. C., Kappelle L. J., van Ginneken B., Velthuis B. K., and Prokop M. Timing-invariant imaging of collateral vessels in acute ischemic stroke. *Stroke*, 44:2194–2199, 2013.

[24] Powers W. J., Derdeyn C. P., Biller J., Coffey C. S., Hoh B. L., Jauch E. C., Johnston K. C., Johnston S. C., Khalessi A. A., Kidwell C. S., Meschia J. F., Ovbiagele B., and Yavagal D. R. 2015 American heart association/American stroke association focused update of the 2013 guidelines for the early management of patients with acute ischemic stroke regarding endovascular treatment. *Stroke*, 46(10):3020–3035, 2015.

[25] Nogueira R. G., Jadhav A. P., Haussen D. C., Bonafe A., Budzik R. F., Bhuva P., Yavagal D. R., Ribo M., Cognard C., Hanel R. A., Sila C. A., Hassan A. E., Millan M., Levy E. I., Mitchell P., Chen M., English J. D., Shah Q. A., Silver F. L., Pereira V. M., Mehta B. P., Baxter B. W., Abraham M. G., Cardona P., Veznedaroglu E., Hellinger F. R., Feng L., Kirmani J. F., Lopes D. K., Jankowitz B. T., Frankel M. R., Costalat V., Vora N. A., Yoo A. J., Malik A. M., Furlan A. J., Rubiera M., Aghaebrahim A., Olivot J.-M., Tekle W. G., Shields R., Graves T., Lewis R. J., Smith W. S., Liebeskind D. S., Saver J. L., and Jovin T. G. Thrombectomy 6 to 24 hours after stroke with a mismatch between deficit and infarct. *N. Engl. J. Med.*, 378(1), 2017.

[26] Chen H., Wu L., Dou Q., Qin J., Li S., Cheng J. Z., Ni D., and Heng P. A. Ultrasound standard plane detection using a composite neural network framework. *IEEE Transactions on Cybernetics*, 47(6):1576–1586, 2017.

[27] Quaday K. A., Salzman J. G., and Gordon B. D. Magnetic resonance imaging and computed tomography utilization trends in an academic ED. *Am J Emerg Med*, 32(6):524–528, 2014.

[28] Talwalkar A. and Uddin S. Trends in emergency department visits for ischemic stroke and transient ischemic attack: United States, 2001-2011. *NCHS data brief*, pages 1–8, 2015.

[29] Meijs M., Pegge S., Prokop M., van Ginneken B., Meijer F. J. A., and Manniesing R. Detection of vessel occlusion in acute stroke is facilitated by color-coded 4D-CTA. In *Eur. Congr. of Radiol.*, 2017.

[30] Manniesing R., Oei M. T., Oostveen L. J., Melendez J., Smit E. J., Platel B., Sánchez C. I., Meijer F. J., Prokop M., and van Ginneken B. White matter and gray matter segmentation in 4D computed tomography. *Nature Scientific Reports*, 7(119), 2017.

[31] Litjens G. J. S., Kooi T., Bejnordi B. E., Setio A. A. A., Ciompi F., Ghafoorian M., van der Laak J. A. W. M., van Ginneken B., and Sánchez C. I. A survey on deep learning in medical image analysis. *Med. Image Anal.*, 42:60–88, 2017.

[32] Moeskops P., Viergever M. A., Mendrik A. M., de Vries L. S., Benders M. J., and Išgum I. Automatic segmentation of MR brain images with a convolutional neural network. *IEEE Trans Med. Imag.*, 35(5):1252–1261, 2016.

[33] Ciompi F., de Hoop B., van Riel S. J., Chung K., Scholten E. T., Oudkerk M., de Jong P. A., Prokop M., and van Ginneken B. Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. *Med. Image Anal.*, 26(1):195–202, 2015.

[34] Setio A. A. A., Ciompi F., Litjens G. J. S., Gerke P., Jacobs C., van Riel S. J., Wille M. M. W., Naqibullah M., Sánchez C. I., and van Ginneken B. Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks. *IEEE Trans Med. Imag.*, 35(5): 1160–1169, 2016.

[35] Ghafoorian M., Karssemeijer N., Heskes T., van Uder I. W. M., de Leeuw F. E., Marchiori E., van Ginneken B., and Platel B. Non-uniform patch sampling with deep convolutional neural networks for white matter hyperintensity segmentation. In *IEEE Int. Symp. Biomed. Imaging*, pages 1414–1417, 2016.

[36] Zhao L. and Jia K. Multiscale CNNs for brain tumor segmentation and diagnosis. *Comput. Math. Methods Med.*, 2016, 2016.

[37] Shakeri M., Tsogkas S., Ferrante E., Lippe S., Kadoury S., Paragios N., and Kokkinos I. Sub-cortical brain structure segmentation using F-CNN's. In *IEEE Int. Symp. Biomed. Imaging*, pages 269–272, 2016.

[38] Song Y., Zhang L., Chen S., Ni D., Lei B., and Wang T. Accurate segmentation of cervical cytoplasm and nuclei based on multiscale convolutional network and graph partitioning. *IEEE Trans. Biomed. Eng.*, 62(10):2421–2433, 2015.

[39] Kamnitsas K., Ledig C., Newcombe V. F., Simpson J. P., Kane A. D., Menon D. K., Rueckert D., and Glocker B. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Med. Image Anal.*, 36:61–78, 2017.

[40] Çiçek Ö., Abdulkadir A., Lienkamp S. S., Brox T., and Ronneberger O. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. *arXiv preprint arXiv:1606.06650*, 2016.

[41] Milletari F., Navab N., and Ahmadi S. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. *arXiv preprint arXiv:1606.04797*, 2016.

[42] Dou Q., Chen H., Yu L., Zhao L., Qin J., Wang D., Mok V. C., Shi L., and Heng P.-A. Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. *IEEE Trans Med. Imag.*, 35(5):1182–1195, 2016.

[43] Brosch T., Tang L. Y., Yoo Y., Li D. K., Traboulsee A., and Tam R. Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation. *IEEE Trans Med. Imag.*, 35(5):1229–1239, 2016.

[44] Korez R., Likar B., Pernuš F., and Vrtovec T. Model-based segmentation of vertebral bodies from mr images with 3D CNNs. In *Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, pages 433–441. Springer, 2016.

[45] Stollenga M. F., Byeon W., Liwicki M., and Schmidhuber J. Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation. *arXiv preprint arXiv:1506.07452*, 2015.

[46] Xie Y., Zhang Z., Sapkota M., and Yang L. Spatial clockwork recurrent neural network for muscle perimysium segmentation. In *Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, volume 9901, pages 185–193. NIH Public Access, 2016.

[47] Poudel R. P., Lamata P., and Montana G. Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation. *arXiv preprint arXiv:1608.03974*, 2016.

[48] Chen J., Yang L., Zhang Y., Alber M. S., and Chen D. Z. Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation. *arXiv preprint arXiv:1609.01006*, 2016.

[49] Chen H., Dou Q., Wang X., Qin J., Cheng J. C., and Heng P.-A. 3D fully convolutional networks for intervertebral disc localization and segmentation. In *Int. Conf. on Med. Imag. and Virt. Real.*, pages 375–382. Springer, 2016.

[50] Shin H.-C., Orton M. R., Collins D. J., Doran S. J., and Leach M. O. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1930–1943, 2013.

[51] Ronneberger O., Fischer P., and Brox T. U-Net: Convolutional networks for biomedical image segmentation. In *Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, pages 234–241. Springer, LNCS, 2015.

[52] Drozdzal M., Vorontsov E., Chartrand G., Kadoury S., and Pal C. The importance of skip connections in biomedical image segmentation. *arXiv preprint arXiv:1608.04117*, 2016.

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

[54] Graham B. Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070*, 2014.

[55] Odena A., Dumoulin V., and Olah C. Deconvolution and checkerboard artifacts. *Distill*, 1(10): e3, 2016.

[56] Nesterov Y. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Dokl.*, 27:372–376, 1983.

[57] He K., Zhang X., Ren S., and Sun J. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[58] Theano Development Team. Theano: a Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

[59] Dieleman S., Schlüter J., Raffel C., Olson E., Sønderby S. K., Nouri D., Maturana D., Thoma M., Battenberg E., Kelly J., et al. Lasagne: First release. *Zenodo: Geneva, Switzerland*, August 2015.

[60] Klein S., Staring M., Murphy K., Viergever M. A., and Pluim J. P. Elastix: a toolbox for intensity-based medical image registration. *IEEE Trans Med. Imag.*, 29(1):196–205, 2010.

[61] Prokop M., Galanski M., and Schaefer-Prokop C. *Spiral and multislice computed tomography of the body*. Thieme Medical Publisher, 2003. ISBN 9780865778702.

[62] van de Leemput S., Meijer F. J. A., Prokop M., and Manniesing R. Cerebral white matter, gray matter and cerebrospinal fluid segmentation in ct using vcast: a volumetric cluster annotation and segmentation tool. In *European Congress of Radiology*, 2017.

[63] Meijs M., Patel A., van de Leemput S. C., Prokop M., Dijk E. J., de Leeuw F., Meijer F. J., Ginneken B., and Manniesing R. Robust segmentation of the full cerebral vasculature in 4D CT of suspected stroke patients. *Sci. Rep.*, 7(1):15622, 2017.

[64] Dice L. R. Measures of the amount of ecologic association between species. *Ecology*, 26(3): 297–302, 1945.

[65] Glorot X. and Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In *Proc. of the 13e Int. Conf. on A.I. and Stats.*, pages 249–256. PMLR, May 2010.

[66] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S. E., Anguelov D., Erhan D., Vanhoucke V., and Rabinovich A. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[67] Powers W. J., Rabinstein A. A., Ackerson T., Adeoye O. M., Bambakidis N. C., Becker K., et al. 2018 guidelines for the early management of patients with acute ischemic stroke: a guideline for healthcare professionals from the american heart association/american stroke association. *Stroke*, 49(3):e46–e99, 2018.

[68] Berkhemer O. A., Fransen P. S., Beumer D., van den Berg L. A., Lingsma H. F., Yoo A. J., et al. A randomized trial of intraarterial treatment for acute ischemic stroke. *N. Engl. J. Med.*, 372(1): 11–20, 2015.

[69] Goyal M., Demchuk A. M., Menon B. K., Eesa M., Rempel J. L., Thornton J., et al. Randomized assessment of rapid endovascular treatment of ischemic stroke. *N. Engl. J. Med.*, 372(11): 1019–1030, 2015.

[70] Jovin T. G., Chamorro A., Cobo E., de Miquel M. A., Molina C. A., Rovira A., et al. Thrombectomy within 8 hours after symptom onset in ischemic stroke. *N. Engl. J. Med.*, 372(24):2296–2306, 2015.

[71] Saver J. L., Goyal M., Bonafe A., Diener H.-C., Levy E. I., Pereira V. M., et al. Stent-retriever thrombectomy after intravenous t-pa vs. t-pa alone in stroke. *N. Engl. J. Med.*, 372(24):2285–2295, 2015.

[72] Campbell B. C., Mitchell P. J., Kleinig T. J., Dewey H. M., Churilov L., Yassi N., et al. Endovascular therapy for ischemic stroke with perfusion-imaging selection. *N. Engl. J. Med.*, 372(11): 1009–1018, 2015.

[73] Nogueira R. G., Jadhav A. P., Haussen D. C., Bonafe A., Budzik R. F., Bhuva P., et al. Thrombectomy 6 to 24 hours after stroke with a mismatch between deficit and infarct. *N. Engl. J. Med.*, 378(1):11–21, 2018.

[74] Albers G. W., Marks M. P., Kemp S., Christensen S., Tsai J. P., Ortega-Gutierrez S., et al. Thrombectomy for stroke at 6 to 16 hours with selection by perfusion imaging. *N. Engl. J. Med.*, 378(8):708–718, 2018.

[75] Ma H., Campbell B. C., Parsons M. W., Churilov L., Levi C. R., Hsu C., et al. Thrombolysis guided by perfusion imaging up to 9 hours after onset of stroke. *N. Engl. J. Med.*, 380(19): 1795–1803, 2019.

[76] Mozaffarian D., Benjamin E. J., Go A. S., Arnett D. K., Blaha M. J., Cushman M., et al. Heart disease and stroke statistics—2016 update: a report from the american heart association. *Circulation*, 133(4):e38–e360, 2016.

[77] Mettler Jr F. A., Huda W., Yoshizumi T. T., and Mahesh M. Effective doses in radiology and diagnostic nuclear medicine: a catalog. *Radiology*, 248(1):254–263, 2008.

[78] Mnyusiwalla A., Aviv R. I., and Symons S. P. Radiation dose from multidetector row ct imaging for acute stroke. *Neuroradiology*, 51(10):635–640, 2009.

[79] Siebert E., Bohner G., Dewey M., Masuhr F., Hoffmann K., Mews J., et al. 320-slice ct neuroimaging: initial clinical experience and image quality evaluation. *Br. J. Radiol.*, 82(979):561–570, 2009.

[80] Manniesing R., Oei M. T., van Ginneken B., and Prokop M. Quantitative dose dependency analysis of whole-brain ct perfusion imaging. *Radiology*, 278(1):190–197, 2015.

[81] Xingjian S., Chen Z., Wang H., Yeung D.-Y., Wong W.-k., and Woo W.-c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Adv. Neural Inf. Process. Syst.*, pages 802–810, 2015.

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

[83] LeCun Y., Bottou L., Bengio Y., and Haffner P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[84] Chong Y. S. and Tay Y. H. Abnormal event detection in videos using spatiotemporal autoencoder. In *International Symposium on Neural Networks*, pages 189–196. Springer, 2017.

[85] Mahjourian R., Wicke M., and Angelova A. Geometry-based next frame prediction from monocular video. In *Intelligent Vehicles Symposium (IV)*, pages 1700–1707. IEEE, 2017.

[86] Luo Y., Ren J., Wang Z., Sun W., Pan J., Liu J., Pang J., and Lin L. LSTM pose machines. *arXiv preprint arXiv:1712.06316*, 2017.

[87] Pigou L., van den Oord A., Dieleman S., Van Herreweghe M., and Dambre J. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *Int. J. Comput. Vis.*, 126(2–4):430–439, 2018.

[88] Zhu G., Zhang L., Shen P., and Song J. Multimodal gesture recognition using 3-D convolution and convolutional LSTM. *IEEE Access*, 5:4517–4524, 2017.

[89] Xu L., Chen X., Cao S., Zhang X., and Chen X. Feasibility study of advanced neural networks applied to sEMG-based force estimation. *Sensors*, 18(10), Sep 2018.

[90] Wu Z., Guo Y., Lin W., Yu S., and Ji Y. A weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems. *Sensors*, 18(4), Apr 2018.

[91] Bilgera C., Yamamoto A., Sawano M., Matsukura H., and Ishida H. Application of convolutional long short-term memory neural networks to signals collected from a sensor network for autonomous gas source localization in outdoor environments. *Sensors*, 18(12), Dec 2018.

[92] Nait Aicha A., Englebienne G., van Schooten K. S., Pijnappels M., and Krose B. Deep learning to predict falls in older adults based on daily-life trunk accelerometry. *Sensors*, 18(5), May 2018.

[93] Hanson J., Paliwal K., Litfin T., Yang Y., and Zhou Y. Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. *Bioinformatics*, 34(23):4039–4045, Dec 2018.

[94] Guo Y., Wang B., Li W., and Yang B. Protein secondary structure prediction improved by recurrent neural networks integrated with two-dimensional convolutional neural networks. *J. Bioinform. Comput. Biol.*, 16(5):1850021, Oct 2018.

[95] Novikov A. A., Major D., Wimmer M., Lenis D., and Buhler K. Deep sequential segmentation of organs in volumetric medical scans. *IEEE Trans. Med. Imag.*, Nov 2018.

[96] Chen Y., Shi B., Wang Z., Sun T., Smith C. D., and Liu J. Accurate and consistent hippocampus segmentation through convolutional LSTM and view ensemble. In *International Workshop on Machine Learning in Medical Imaging*, pages 88–96. Springer, 2017.

[97] Bao S., Wang P., Mok T. C. W., and Chung A. C. S. 3D randomized connection network with graph-based label inference. *IEEE Trans. Image Process.*, Apr 2018.

[98] Xu W. and LeBeau J. M. A deep convolutional neural network to analyze position averaged convergent beam electron diffraction patterns. *Ultramicroscopy*, 188:59–69, 05 2018.

[99] Vargas J., Spiotta A., and Chatterjee A. R. Initial Experiences with Artificial Neural Networks in Detection of CT Perfusion Deficits. *World Neurosurg.*, Oct 2018.

[100] McCann M. T., Jin K. H., and Unser M. A review of convolutional neural networks for inverse problems in imaging. *arXiv preprint arXiv:1710.04011*, 2017.

[101] Wang G., Ye J. C., Mueller K., and Fessler J. A. Image reconstruction is a new frontier of machine learning. *IEEE Trans. Med. Imag.*, 37(6):1289–1296, 2018.

[102] Nie D., Cao X., Gao Y., Wang L., and Shen D. Estimating CT image from MRI data using 3D fully convolutional networks. In *Deep Learning and Data Labeling for Medical Applications*, pages 170–178. Springer, Cham, 2016. ISBN 978-3-319-46976-8.

[103] Bahrami K., Shi F., Rekik I., and Shen D. Convolutional neural network for reconstruction of 7T-like images from 3T MRI using appearance and anatomical features. In *Deep Learning and Data Labeling for Medical Applications*, pages 39–47. Springer, Cham, 2016. ISBN 978-3-319-46976-8.

[104] Chen H., Zhang Y., Zhang W., Liao P., Li K., Zhou J., and Wang G. Low-dose CT via convolutional neural network. *Biomed. Opt. Express*, 8(2):679–694, 2017.

[105] Kang E., Min J., and Ye J. C. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Med. Phys.*, 44(10), 2017.

[106] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., et al. Generative adversarial nets. In *Adv. Neural Inf. Process. Syst.*, pages 2672–2680, 2014.

[107] Schawinski K., Zhang C., Zhang H., Fowler L., and Santhanam G. K. Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit. *Mon. Notices Royal Astron. Soc.*, 467(1):L110–L114, 2017.

[108] Ledig C., Theis L., Huszár F., Caballero J., Cunningham A., Acosta A., et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[109] Zhu J.-Y., Park T., Isola P., and Efros A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

[110] Wolterink J. M., Dinkla A. M., Savenije M. H. F., Seevinck P. R., van den Berg C. A., and Išgum I. Deep MR to CT synthesis using unpaired data. In *International Workshop on Simulation and Synthesis in Medical Imaging*, pages 14–23. Springer, 2017.

[111] van de Leemput S. C., Prokop M., van Ginneken B., and Manniesing R. Stacked bidirectional convolutional LSTMs for 3D non-contrast CT reconstruction from spatiotemporal 4D CT. In *Medical Imaging with Deep Learning*, 2018.

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

[113] Jozefowicz R., Zaremba W., and Sutskever I. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.

[114] Chollet F. et al. Keras. `https://github.com/fchollet/keras`, 2015.

[115] Wang Z., Bovik A. C., Sheikh H. R., Simoncelli E. P., et al. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[116] Bai S., Kolter J. Z., and Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[117] van de Leemput S. C., Patel A., and Manniesing R. Full volumetric brain tissue segmentation in non-contrast CT using memory efficient convolutional LSTMs. In *Medical Imaging meets NeurIPS*, 2018.

[118] Chen T., Xu B., Zhang C., and Guestrin C. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[119] He K., Zhang X., Ren S., and Sun J. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[120] Gomez A. N., Ren M., Urtasun R., and Grosse R. B. The reversible residual network: Backpropagation without storing activations. *arXiv preprint arXiv:1707.04585*, 2017.

[121] Chang B., Meng L., Haber E., Ruthotto L., Begert D., and Holtham E. Reversible architectures for arbitrarily deep residual neural networks. *arXiv preprint arXiv:1709.03698*, 2017.

[122] Jacobsen J.-H., Smeulders A., and Oyallon E. i-RevNet: Deep invertible networks. In *ICLR*, 2018.

[123] Krizhevsky A. and Hinton G. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 1(4):7, 2009.

[124] Dinh L., Krueger D., and Bengio Y. NICE: non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[125] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G. S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Y.Jia, Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mané D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viégas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., and Zheng X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[126] Paszke A., Gross S., Chintala S., Chanan G., Yang E., DeVito Z., Lin Z., Desmaison A., Antiga L., and Lerer A. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[127] Dinh L., Sohl-Dickstein J., and Bengio S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[128] Kingma D. P. and Dhariwal P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[129] Martens J. and Sutskever I. Training deep and recurrent networks with hessian-free optimization. In *Neural networks: Tricks of the trade*, pages 479–535. Springer, 2012.

[130] Ouderaa T. F. v. d. and Worrall D. E. Reversible gans for memory-efficient image-to-image translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[131] van der Ouderaa T. F., Worrall D. E., and van Ginneken B. Chest CT super-resolution and domain-adaptation using memory-efficient 3d reversible GANs. In *International Conference on Medical Imaging with Deep Learning*, London, United Kingdom, 08–10 Jul 2019.

[132] Powers W. J., Rabinstein A. A., Ackerson T., Adeoye O. M., Bambakidis N. C., Becker K., Biller J., Brown M., Demaerschalk B. M., Hoh B., Jauch E. C., Kidwell C. S., Leslie-Mazwi T. M., Ovbiagele B., Scott P. A., Sheth K. N., Southerland A. M., Summers D. V., Tirschwell D. L., and null null. Guidelines for the early management of patients with acute ischemic stroke: 2019 update to the 2018 guidelines for the early management of acute ischemic stroke: A guideline for healthcare professionals from the american heart association/american stroke association. *Stroke*, 50(12):e344–e418, 2019.

[133] Olthuis S. G. H., Pirson F. A. V., Pinckaers F. M. E., Hinsenveld W. H., Nieboer D., Ceulemans A., Knapen R. R. M. M., Robbe M. M. Q., Berkhemer O. A., van Walderveen M. A. A., Lycklama à Nijeholt G. J., Uyttenboogaart M., Schonewille W. J., van der Sluijs P. M., Wolff L., van Voorst H., Postma A. A., Roosendaal S. D., van der Hoorn A., Emmer B. J., Krietemeijer M. G. M., van Doormaal P.-J., Roozenbeek B., Goldhoorn R.-J. B., Staals J., de Ridder I. R., van der Leij C., Coutinho J. M., van der Worp H. B., Lo R. T. H., Bokkers R. P. H., van Dijk E. I., Boogaarts H. D., Wermer M. J. H., van Es A. C. G. M., van Tuijl J. H., Kortman H. G. J., Gons R. A. R., Yo L. S. F., Vos J.-A., de Laat K. F., van Dijk L. C., van den Wijngaard I. R., Hofmeijer J., Martens J. M., Brouwers P. J. A. M., Bulut T., Remmers M. J. M., de Jong T. E. A. M., den Hertog H. M., van Hasselt B. A. A. M., Rozeman A. D., Elgersma O. E. H., van der Veen B., Sudiono D. R., Lingsma H. F., Roos Y. B. W. E. M., Majoie C. B. L. M., van der Lugt A., Dippel D. W. J., van Zwam W. H., and van Oostenbrugge R. J. Endovascular treatment versus no endovascular treatment after 6–24 h in patients with ischaemic stroke and collateral flow on ct angiography (mr clean-late) in the netherlands: a multicentre, open-label, blinded-endpoint, randomised, controlled, phase 3 trial. *The Lancet*, 401(10385):1371–1380, Apr 2023.

[134] Nguyen T. N., Castonguay A. C., Siegler J. E., Nagel S., Lansberg M. G., de Havenon A., Sheth S. A., Abdalkader M., Tsai J. P., Albers G. W., et al. Mechanical thrombectomy in the late presentation of anterior circulation large vessel occlusion stroke: a guideline from the society of vascular and interventional neurology guidelines and practice standards committee. *Stroke: Vascular and Interventional Neurology*, 3(1):e000512, 2023.

[135] Sarraj A., Hassan A. E., Abraham M. G., Ortega-Gutierrez S., Kasner S. E., Hussain M. S., Chen M., Blackburn S., Sitton C. W., Churilov L., et al. Trial of endovascular thrombectomy for large ischemic strokes. *New England Journal of Medicine*, 388(14):1259–1271, 2023.

[136] Wuschner A. E., Flakus M. J., Wallat E. M., Reinhardt J. M., Shanmuganayagam D., Christensen G. E., Gerard S. E., and Bayouth J. E. Ct-derived vessel segmentation for analysis of post-radiation therapy changes in vasculature and perfusion. *Frontiers in physiology*, page 2226, 2022.

[137] Chen C., Bivard A., Lin L., Levi C. R., Spratt N. J., and Parsons M. W. Thresholds for infarction vary between gray matter and white matter in acute ischemic stroke: a ct perfusion study. *Journal of Cerebral Blood Flow & Metabolism*, 39(3):536–546, 2019.

[138] Meijs M., Pegge S., Murayama K., Boogaarts H., Prokop M., Willems P., Manniesing R., and Meijer F. Color-mapping of 4d-cta for the detection of cranial arteriovenous shunts. *American Journal of Neuroradiology*, 40(9):1498–1504, 2019.

[139] Demeestere J., Wouters A., Christensen S., Lemmens R., and Lansberg M. G. Review of perfusion imaging in acute ischemic stroke: from time to tissue. *Stroke*, 51(3):1017–1024, 2020.

[140] Rubiera M., Garcia-Tornel A., Olivé-Gadea M., Campos D., Requena M., Vert C., Pagola J., Rodriguez-Luna D., Muchada M., Boned S., Rodriguez-Villatoro N., Juega J., Deck M., Sanjuan E., Hernandez D., Piñana C., Tomasello A., Molina C. A., and Ribo M. Computed tomography perfusion after thrombectomy. *Stroke*, 51(6):1736–1742, 2020.

[141] Boers A. M., Jansen I. G., Brown S., Lingsma H. F., Beenen L. F., Devlin T. G., San Román L., Heo J.-H., Ribó M., Almekhlafi M. A., et al. Mediation of the relationship between endovascular therapy and functional outcome by follow-up infarct volume in patients with acute ischemic stroke. *JAMA neurology*, 76(2):194–202, 2019.

[142] Compagne K. C., Boers A., Marquering H. A., Berkhemer O. A., Yoo A. J., Beenen L. F., van Oostenbrugge R., van Zwam W., Roos Y., Majoie C., et al. Follow-up infarct volume as a mediator of endovascular treatment effect on functional outcome in ischaemic stroke. *European radiology*, 29:736–744, 2019.

[143] Hill M. D., Warach S., and Rostanski S. K. Should primary stroke centers perform advanced imaging? *Stroke*, 53(4):1423–1430, 2022.

[144] Menon B. K., d'Esterre C. D., Qazi E. M., Almekhlafi M., Hahn L., Demchuk A. M., and Goyal M. Multiphase ct angiography: a new tool for the imaging triage of patients with acute ischemic stroke. *Radiology*, 275(2):510–520, 2015.

[145] Amy Y., Zerna C., Assis Z., Holodinsky J. K., Randhawa P. A., Najm M., Goyal M., Menon B. K., Demchuk A. M., Coutts S. B., et al. Multiphase ct angiography increases detection of anterior circulation intracranial occlusion. *Neurology*, 87(6):609–616, 2016.

[146] Dundamadappa S., Iyer K., Agrawal A., and Choi D. Multiphase ct angiography: a useful technique in acute stroke imaging—collaterals and beyond. *American Journal of Neuroradiology*, 42(2):221–227, 2021.

[147] Oei M. T., Meijer F. J., van der Woude W.-J., Smit E. J., van Ginneken B., Manniesing R., and Prokop M. Interleaving cerebral ct perfusion with neck ct angiography. part ii: clinical implementation and image quality. *European Radiology*, 27:2411–2418, 2017.

[148] Oei M. T., Meijer F. J., van der Woude W.-J., Smit E. J., van Ginneken B., Prokop M., and Manniesing R. Interleaving cerebral ct perfusion with neck ct angiography part i. proof of concept and accuracy of cerebral perfusion values. *European Radiology*, 27:2649–2656, 2017.

[149] Wafa H. A., Wolfe C. D., Emmett E., Roth G. A., Johnson C. O., and Wang Y. Burden of stroke in europe: thirty-year projections of incidence, prevalence, deaths, and disability-adjusted life years. *Stroke*, 51(8):2418–2427, 2020.

[150] Ovbiagele B., Goldstein L. B., Higashida R. T., Howard V. J., Johnston S. C., Khavjou O. A., Lackland D. T., Lichtman J. H., Mohl S., Sacco R. L., et al. Forecasting the future of stroke in the united states: a policy statement from the american heart association and american stroke association. *Stroke*, 44(8):2361–2375, 2013.

[151] Isensee F., Jaeger P. F., Kohl S. A., Petersen J., and Maier-Hein K. H. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

[152] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., and Polosukhin I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Dankwoord

Allereerst zou ik graag mijn promotoren en mijn co-promotor willen bedanken voor hun begeleiding gedurende mijn promotietraject. **Bram van Ginneken**, bedankt voor al je advies en je kritische blik gedurende al onze discussies. In het bijzonder wil ik je bedanken voor de begeleiding tijdens de afronding van dit proefschrift. Het is heel waardevol om iemand te hebben zoals jij die helpt met het stellen van realistische doelen om zo te zorgen voor een goede afronding. **Mathias Prokop**, bedankt voor al je suggesties, enthousiasme en je toewijding. Je was altijd in staat om mij te motiveren met je enthousiasme en onze onderzoeksgroep te voorzien van een frisse blik. **Rashindra Manniesing**, ik wil jou als mijn dagelijkse begeleider in het bijzonder bedanken voor alle begeleiding en onze vriendschap. Met jouw enthousiasme en inzet wist je mij altijd te motiveren en creëerde je een prettige werksfeer. Ik heb ook altijd erg genoten van onze discussies over deep learning toepassingen. Bedankt voor de leuke tijd in de 4DCT/neuro groep.

Ik wil graag al mijn co-authors bedanken die hebben bijgedragen aan de hoofdstukken in dit proefschrift. **Anton Meijer**, bedankt voor je hulp en expertise tijdens het realiseren van hoofdstuk 2. Je was altijd bereid om onze klinische vragen te beantwoorden en te duiden wat we zagen op de vele scans. Jouw hulp en inzet waren cruciaal voor het realiseren van onze database en het mogelijk maken van onze projecten. **Jonas**, bedankt voor je hulp en inzet voor het maken van hoofdstuk 4. Niet alleen hebben we regelmatig overleg gehad over technische implementaties van neurale netwerken, maar we zijn ook tweemaal samen naar Canada geweest voor de ICLR en de NeurIPS converenties. Ik heb daar ontzettend genoten van onze gesprekken, het ‘gore‘ eten, en het verkennen van Vancouver en Montréal. Ook wil ik graag **Lisan Kaal** en **Loes Vos** bedanken voor hun inzet tijdens het intensieve annotatiewerk van de hersengebieden van de 4D CT data.

Gedurende mijn promotietraject heb ik het voorrecht gehad om nauw samen te mogen werken met **Midas** en **Ajay**, de andere leden van het 4DCT-team. Samen waren wij de Cerebros. Ik wil jullie graag allebei ontzettend bedanken voor jullie hulp en de goede werksfeer. Jullie hebben gezorgd voor een goede balans tussen werk en de benodigde ontspanning en hielpen mij altijd om de klinische relevantie niet uit het oog te verliezen. Midas, ik wil je graag bedanken voor jouw praktische insteek en je scherpe analyse van problemen, waardoor je mij vaak hielp om de focus te behouden. Ajay, bedankt voor je geduld tijdens het beantwoorden van al mijn klinische vragen en voor het feit dat ik altijd bij je terecht kon om mijn problemen te bespreken.

Mede dankzij jullie heb ik mijn promotietraject in die vier jaar altijd als heel prettig ervaren. Ik ben dan ook vereerd dat jullie allebei mijn paranymphen willen zijn.
Alle andere collega's van DIAG wil ik natuurlijk heel erg bedanken, voor de gesprekken, de gezellige tijd in de groep, de borrels en etentjes, en de leuke tijd bij conferenties **Thomas, Freerk, Mohsen, Babak, Bart, Cristina, Suzan, Kevin, Bram Platel, Clarisa, Henkjan, Nico, Geert, Francesco, Sven, Nikita, Alejandro, Sjoerd, Charlotte, Jan-Jurre, Rick, Albert, Wendelien, Mark, Kaman, Arnoud, Gabriel, Anton, Ecem, Thijs, David, Dagmar**, en **John-Melle**. Ik wil graag **Colin** bedanken voor het begrip en de ruimte die je me liet voor het afronden van dit proefschrift terwijl ik voor je werkte na het aflopen van mijn promotiecontract. Ik wil ook graag **Miriam, Mike, Anne, Chris, Harm** van het **RSE team** bedanken voor de fijne samenwerking. In het bijzonder wil ik **Paul Gerke** bedanken waarmee ik samen een tijdlang de eerste groepsbrede deep learning image onderhield. Ik geniet altijd van onze gesprekken en gedeelde interesses en heb veel van jou geleerd. **James Meakin**, thank you for all your help for setting up the MemCNN codebase and your suggestion for submitting my MemCNN article to the Journal of Open Source Software.

Aan mijn badmintonvrienden: **Frank, Paul, Edwin, Willem, Will, Gerard, Zia, Eli, Moniek, Saskia, Jacqueline, Bjorn** en vele anderen, bedankt voor alle leuke momenten, wedstrijdjes, en de broodnodige ontspanning en afleiding naast mijn werk en promotie.

Aan mijn schoonfamilie: **Marie-José, Jan, Merten, en Josine**, bedankt voor jullie interesse en steun gedurende al die jaren. Bedankt dat jullie mij altijd hebben geaccepteerd als deel van de familie.

Linde en **Roos**, mijn zusjes, bedankt voor alle gesprekken en alle leuke dingen die we samen hebben gedaan. Het lukt me niet jullie altijd duidelijk te maken wat ik allemaal uitspook en ik hoop dat dit proefschrift daar wat meer inzicht in kan geven. **Jan** en **Tim**, bedankt voor jullie interesse in mijn onderzoek en alle goede gesprekken.

Lieve **moeder**, bedankt voor al je toewijding en liefde al die jaren. Jij hebt mij mijn hele leven laten zien dat volharding in tijden van tegenslag essentieel is. Hier heb ik veel aan gehad tijdens mijn promotie en daar ben ik jou zeer dankbaar voor.

Tot slot wil ik mijn vrouw **Janneke** heel erg bedanken. Bedankt voor jouw geduld tijdens de eindeloze uren die ik tijdens (en na) mijn promotieperiode achter de computer heb doorgebracht om mijn proefschrift af te ronden. Dit bracht natuurlijk extra

last voor jou, maar ondanks alles heb jij mij hierin altijd gesteund. Zonder jou had ik dit nooit kunnen volbrengen en ik ben je daar voor altijd dankbaar voor. Ik hou van jou, en ik hoop dat wij samen nog vele jaren kunnen genieten van elkaar en van onze drie prachtige kinderen: Casper, Esmée, en Philine.

Curriculum vitae

Silvester Christiaan van de Leemput was born in de Haarlemmermeer, the Netherlands, on March 19th, 1987. He received his BSc. and MSc. degree in artificial intelligence from Radboud University, Nijmegen, the Netherlands, in 2013 and 2015, respectively. In August 2015, he started working as a Ph.D. student at the Diagnostic Image Analysis Group at the Radboud university medical center in Nijmegen, the Netherlands. His PhD project focuses on deep learning techniques for imaging in the acute stroke setting, supervised by Rashindra Manniesing, Bram van Ginneken, and Mathias Prokop. The results of these works are described in this thesis. Since 2019 he has been working as a scientific programmer at the Diagnostic Image Analysis Group.

9 789493 296633

Radboudumc university medical center

Radboud University