Deep Learning for lesion and thrombus segmentation from cerebral MRI

Jonathan Kobold

université PARIS-SACLAY

NNT : 2019SACLE044

Thèse de doctorat

Deep Learning for Lesion and Thrombus Segmentation from Cerebral MRI

Apprentissage Profond pour la Segmentation de Lésion et de Thrombus dans des IRM Cérébrales

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université d'Evry-Val-d'Essonne

École doctorale n°580 sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Traitement du signal et des images

Thèse présentée et soutenue à Evry, le 2.12.2019, par

JONATHAN KOBOLD

Composition du Jury :

Pierre-Yves Gumery Pr, Université Grenoble Alpes (TIMC-IMAG)	Président
Jenny Benois-Pineau Pr, Université Bordeaux (LaBRI)	Rapporteur
Rachid Jennane Pr, Université d'Orléans (I3MTO)	Rapporteur
Elmar Lang Pr, Universität Regensburg (CIML)	Examinateur
Said Mammar Pr, Université d'Evry-Val-d'Essonne (IBISC)	Examinateur
Hichem Maaref Pr, Université d'Evry-Val-d'Essonne (IBISC)	Directeur de thèse
Vincent Vigneron MCF-HDR, Université d'Evry-Val-d'Essonne (IBISC)	Co-directeur de thèse
Didier Smadja Pr, Université Paris Sud (INSERM)	Invité
Nicolas Chausson Dr, Université Paris Sud (INSERM)	Invité

Contents

List of Figures

List of Tables

List of Acronyms and Symbols

ADC	Apparent diffusion coefficient
BraTS	Brain tumor segmentation challenge
CHSF	Centre Hospitalier Sud-Francilien
CLSTM	Convolutional long-short time memory
CNN	Convolutional neural network
CSF	Cerebrospinal fluid
CT	Computed tomography
CTA	Computed tomography angiography
DICOM	Data format for clinical images
DWI	Diffusion-weighted imaging
ELU	Exponential linear unit
EPI	Echo planar imaging
FLAIR	Fluid attenuated inversion recovery
GPU	Graphics processing unit
LSTM	Long-short time memory
MCA	Middle carotid artery
MIP	Maximum intensity projections
MRI	Magnetic resonance imaging
NCCT	Non contrast computed tomography
NIHSS	National institutes of health stroke scale
NMR	Nuclear magnetic resonance
NN	Neural network
PACS	Picture archiving system
PWI	Perfusion weighted imaging
RAPID	Rapid processing of perfusion and diffusion
RELU	Rectified linear unit
RF	Radio frequency
RNN	Recurrent neural network
SPM	Statistical parametric mapping, a toolbox for MRI pre-processing
SWAN	Susceptibility weighted angiography
ToF	Time of flight angiography
tPA	Tissue plasminogen activator

T_1	Longitudinal or spin-lattice relaxation time
T_2	Transversal or spin-spin relaxation time
T_2^*	Modified T_2
$\lceil x \rceil$	The ceiling of the number x
$a * b$	The convolution of the functions a and b
Δ	the gradient of a loss function
$\Theta(x)$	The heaviside step function of the number x
W	Weight for a neural network layer
$\sigma(x)$	(Element wise) logistic function of the number or matrix x
$\circ$	Hadamard product
δ	Chemical shift
γ	Gyromagnetic ratio
ω_0	Larmor frequency
$\hbar$	Reduced Plank constant
k_b	Boltzmann constant
$\mathcal{B}$	Boltzmann factor
$E[x]$	The expected value of the random variable x

Danksagung - Remerciements

En premier je souhaite remercier professeur Hichem Maaref qui a pris la charge de directeur de thèse. Je remercie vivement Vincent Vigneron. On ne peut pas avoir un meilleur encadrant. Il m'a toujours donné la liberté et le temps pour poursuivre mes idées, tout en m'aidant à me réorienter si je me retrouvais dans une impasse. Il a fait tous ses efforts pour créer l'environement qui m'a permis de réussir mon projèt de thèse. Egalement il s'est engagé pour l'achat du materiel qui était nécessaire pour la partie "Deep Learning" de ma thèse et il m'a permis de dépasser les aléas de l'administration. Un grand merci aux membres de la jury Jenny Benois-Pineau, Rachid Jennane, Elmar Lang et Pierre-Yves Gumery pour la lecture minutieuse de mon manuscript de thèse et de prendre le temps pour ma soutenance. Je remercie sincèrement l'équipe neurologie du CHSF, Didier Smadja, Nicolas Chausson, Cosmin Aleçu, Manvel Aghasaryan et Yann L'Hermitte pour leur investissement dans notre projet. À travers de nombreuses réunions j'ai pu apprendre des détails minutieux sur les AVCs et sur leur representation sur IRM. L'équipe a consacré un temps considerable dans la création et la segmentation de notre base des données. Un merci tout particulier va à Nidhal Ben Achour, une ancienne députée de l'équipe neurologie, qui a passé tous les lundis après-midis pendant les trois premiers mois de thèse avec moi pour m'apprendre à lire et à comprendre les IRMs. Leur contribution pour le success du projet est inestimable.

Mein größter Dank gebührt meinen Eltern, Gabi und Uwe. Sie haben mir alles Nötige mit auf meinen Weg gegeben und mich immer bei allen meinen Vorhaben unterstützt, den kleinen wie den großen Dingen. Ohne Euch wäre ich nicht so weit gekommen. Auch möchte ich Prof. Dr. Elmar Lang danken, der mir nicht nur die Promotionsstelle vermittelt hat sondern mich auch in zahlreichen Diskussionen inspiriert hat. Auch hat er mich beim Schreiben meiner Veröffentlichungen unterstützt. Ein besonderer Dank geht an Anna Artischuk die mich mit ihrem Humor immer aufgemuntert hat.

Auch hat sie sich nicht gescheut mir wiederholt meine altbackenen Präsentationsfolien vorzuhalten und zu verbessern. Dadurch durfte ich lernen optisch ansprechende Grafiken und Präsentationen zu erstellen.

And last but not least I want to thank Prof. Dr. Ana Maria Tomé who invited me to her lab in Aveiro, Portugal. I was able to spend a month with her and her colleagues in a very friendly environment and could give my first public talk there.

Introduction

It is teatime. Marie, 62, lifts her cup of black tea to her lips when her arm goes suddenly numb. She manages to set down the cup but she has trouble moving her arm. She is afraid, not knowing what is happening to her. But her friend has the presence of mind to call an ambulance immediately. Thirty minutes later at the hospital an MRI reveals the cause: stroke. Luckily Marie arrived quickly at the hospital and can be treated with thrombolysis. She can leave the hospital a few days later and she doesn't retain any permanent damage.

In this way or similar many people experience a stroke every day. Stroke diagnosis is a fascinating combination of medical knowledge and physics. Magnetic resonance imaging (MRI) is the most important real world application of quantum physics which allows us to look into the brain of living people without damaging them in any way. Combined with the medical knowledge of the blood flow in the human brain this allows to identify the thrombus which causes the problems. The MRI also gives away information about the amount of damage the brain has sustained. The doctor now needs to decide for one of four treatment options based on his interpretation of the MRI. This is the point where things become difficult: for one the interpretation of the MRI needs a highly trained specialist and second a stroke requires a treatment decision within minutes. Thus the interpretation of the MRI is really only a quick look at the images by the doctor without the possibility of a thorough evaluation. It is amazing that these specialists can gather enough information for a treatment decision in such a short time. However it has been shown that precise measurements of the lesion volume [1] and thrombus length [2, 3] are good predictors for the patients response to the various treatments.

And this is the starting point of this thesis. The goal is to provide an automatic method for segmenting the lesion and the thrombus. This will improve the interpretation of stroke MRI by giving numerical values for the

relevant parameters instead of the rough guesses which are currently used. The methods of choice for this kind of segmentation problem are machine learning methods. This makes this thesis an intriguing subject combining knowledge from the fields of physics, machine learning and medicine. This multidisciplinary approach is also reflected in the structure of this thesis by explaining the basics from each discipline separately. Of course it is indispensable to understand what stroke is and how it works. This is explained in the medical part (section 1.1). To understand the MR images it is necessary to understand how they are generated, *i.e.* the physics of MRI (section 1.2.1). The medical and physical knowledge are merged in the interpretation of the MR images (section 1.2). Then machine learning and its important field deep learning are introduced in chapter 2. Together with a review of existing automatic methods for stroke MRI (section 1.3) this is the basis for exploring the stroke segmentation problem.

One of the big opportunities of this thesis is that it not only allows to apply machine learning to a real problem but also the problem is so complex that it allows new insights into machine learning theory. New approaches on neural network design are elaborated in chapter 3 and their application on the stroke segmentation task is detailed in chapter 4. Altogether the subject of this thesis is very complex and encompasses three disciplines, physics, medicine and machine learning. The latter has not seen any clinical application in a stroke scenario yet. The complexity of the problems entails the further exploration of machine learning theory. And the results can be directly applied to a relevant problem of today's medicine and will eventually find its way to clinical practice. Most thesis subjects are either of theoretical or applied nature and it is the special beauty of the stroke MRI segmentation problem that it allows to have both.

Chapter 1

Stroke Diagnosis

1.1 Stroke

Stroke is a disease which affects the brain. It is one of the most lethal diseases, responsible for around 22% of all deaths world-wide[1]. Therefore being the number 2 cause of death after heart attacks regardless of social status. Those who survive stroke are often left with major disabilities, requiring expensive subsequent care. So it is not a surprise that stroke is subject to extensive research.

There are three different types of stroke which share the same symptoms:

- Haemorrhagic Stroke
- Transient Ischaemic Attack
- Ischaemic Stroke

The first one is due to haemorrhage in the brain. The mechanisms leading to haemorrhagic stroke are quite different to the other two and will not be discussed here. This work is concerned with ischaemic stroke which is caused by a lack of oxygen. The transient ischaemic attack shares the same mechanism with the ischaemic stroke, with the difference that the cause dissolves by itself and doesn't need treatment. Therefore they are also called warning strokes, as they indicate a elevated risk for stroke. In the following the word stroke will always mean an ischaemic stroke.

[1] http://www.who.int/mediacentre/factsheets/fs310/en/

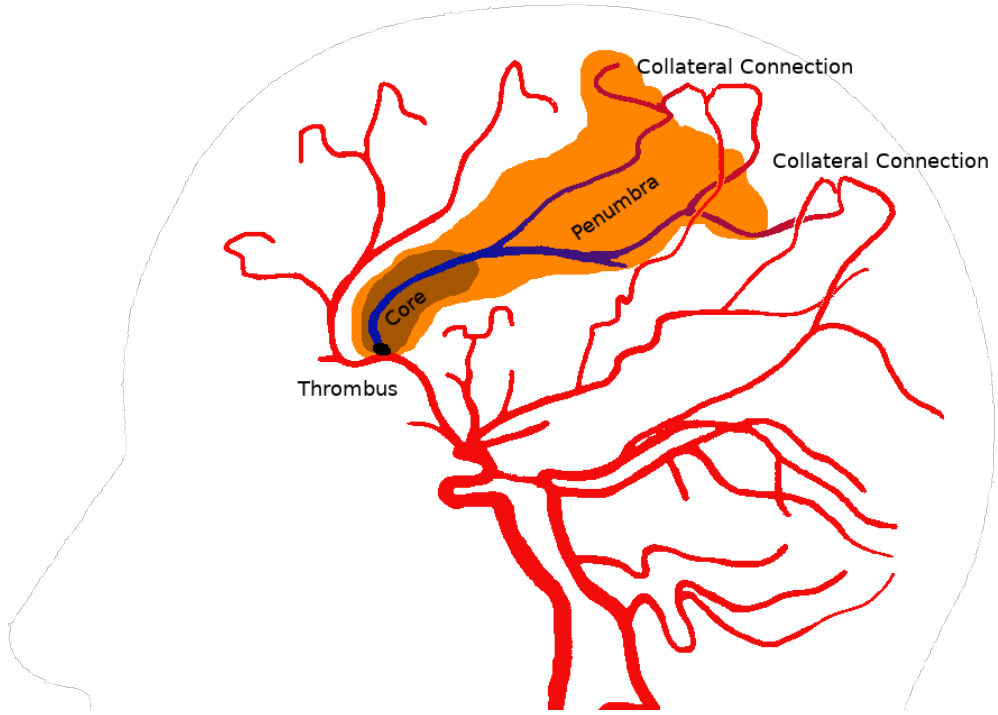


Figure 1.1 Schematic representation of an ischaemic stroke. The black dot represents the thrombus. Red lines are normal arteries and blue lines are arteries with restricted blood flow. The infarct is brown and the penumbra is orange. Collateral connections are indicated. Blood vessels and head adapted from [4]

This chapter will at first elaborate the formation of stroke and the resulting symptoms. Then the possible treatments and the decision process leading to a choice of treatment will be outlined.

1.1.1 The Mechanism of Stroke

Ischaemic stroke is the most common type of stroke. It is caused by a blood clot (**thrombus**) which blocks an artery in the brain and the brain tissue supplied by this artery suffers from a lack of oxygen (**ischaemia**). This section looks at the details of this process and the consequences of ischaemia.

There are hundreds of possible causes which can lead to the formation of a blood clot. Asian populations have blood clots for different reasons than European or African populations. Thus the blood clot formation will not be discussed here. One common possibility is stagnant blood, for example a vein which has been cut off by sitting on it for hours during a long flight.

The stagnant blood starts to coagulate and when the person moves again, *i.e.* gets out of the plane, the coagulated blood makes its way to the brain in the form of a blood clot.

The arterial tree of the brain consist of branches with ever decreasing diameters who eventually become capillary vessels. These will then join again in the venous system. Once a thrombus arrives in the brain he will become stuck, when the artery diameter decreases (figure 1.1, black dot). This can occur as a complete occlusion (no blood can flow past the thrombus) or a partial occlusion (still some blood can pass the thrombus).

The area supplied by the blocked artery is then suddenly blocked off its oxygen supply. This is the moment when a rescue mechanism kicks in. The branches of the arterial tree are interconnected by so called collateral arteries [4]. Usually there is no flow through these, because the blood pressure is equal on both sides. Due to the occlusion by the thrombus the pressure drops in the affected artery and blood flows through the collateral arteries, trying to sustain the blood flow (figure 1.1).

But the flow through the collateral arteries is not enough to replace the flow from the main artery completely. How much of the blood flow can be recovered by collateral circulation depends on the individual. Some persons might have few collateral arteries, other a lot [4]. Therefore the oxygen supply in the affected area does not drop to zero, but rather to a severely reduced level depending on the distance to the collateral connections (and the thrombus in the case of a partial occlusion). A region with insufficient blood supply is called an ischaemic region or sometimes hypoperfused tissue.

Due to this under-supply with oxygen the neurons' K^+ channels will stop working [5] to reduce their oxygen consumption in order to stay alive. Osmotic pressure then leads to a massive K^+ outflow from the neurons which causes a depolarisation of the cell membrane. Consequently the neuron becomes electrically inactive. This is the moment when the symptoms start (stroke onset). Whatever the job of the affected region was, this functionality will suddenly be lost. This can be literally anything, from motor skills like walking or speech, sensory systems like sight or touch, to memory.

The under-supplied neurons will eventually die. Without oxygen supply they die within 5 minutes [5]. Luckily even the tiniest amount of oxygen prolongs their life considerably. Stroke patients with ischaemic tissue still living up to 24 hours and more after stroke onset have been reported [6]. As the oxygen distribution is not equal in the ischaemic region, the neurons will gradually

die over time. The region of already dead neurons is called **infarct**[2] or core (figure 1.1, brown region)[7]. The part of the ischaemic region which is still alive is called the **penumbra** (figure 1.1, orange region)[7]. The loss of tissue in the infarct is irreversible. Tissue in the penumbra can be saved and returned to functionality by restoring the blood supply. Restoring blood flow through the infarct can lead to haemorrhages [8], as the artery walls suffer from the ischaemia as well and become weak. The infarct grows with time, until it fills the whole ischaemic region. In turn the penumbra becomes smaller with time until it vanishes.

The exact cell biological process of the transition from healthy to ischaemic to dead is still subject to current research [9, 10, 11]. Here only one aspect will be highlighted, because it is an important effect for a diagnosis technique[3]. As the ischaemia progresses the diffusion coefficient of the free water molecules decreases. There are several hypothesis how this happens [12]. With decreasing oxygen level, the neuron is no longer able to keep up the functionality of its cell membrane (**cytotoxic oedema**). More water may now enter into the cells because of the osmotic pressure. Once the water is inside the cells it has a reduced space for diffusion. Thus the diffusion coefficient of water drops in the regions where the cell membranes dysfunction. Alternatively the swelling of the cells due to the water influx decreases the extra cellular space. So the space for free water diffusion is reduced and therefore the diffusion coefficient. The last hypothesis says that with the cease of the cell function the transport of plasma inside the cell (cytoplasmic streaming) is also stopped. As the transported plasma contains water this is a contribution to the moving water which is measured with the diffusion coefficient. The truth is probably a mix of the three. The decrease of the diffusion coefficient is not yet the point where the cells die (**ionic oedema**), but close.

Usually it takes between 6 and 12 hours for the infarct to grow to full size. At this point some patients die, because they lost critical functionality. For those who survive the infarcted tissue starts to swell after around 24 hours [13]. This compresses adjacent regions of the brain and the pressure in the brain rises. This compression can cause the death of even more tissue, which in turn can lead to further disabilities or the patients death [8].

[2] In fact any tissue which died of ischaemia is called infarct.

[3] Diffusion Weighted Imaging, see 1.2.2

1.1.2 Stroke Treatment

The goal of treating stroke is to achieve the best possible outcome for the patient. Ethical problems will not be considered here[4]. In the scope of this work, the best possible outcome is defined as saving the penumbra which exists when the patient arrives at the hospital, without damaging any other part of the brain.

Patients with elevated stroke risk can be pre-emptively treated with anticoagulants, to prevent the formation of blood clots and therefore reduce their risk to experience a stroke.

For most patients who already have a stroke the restoration of the blood flow is the ultimate goal. Although for some patients this would be ill advised. If the patient arrives so late at the hospital that the infarct has reached its full size, the only treatment option is to consider decompressive therapy to reduce the damage done by the swelling tissue [14]. For small infarcts this might not be necessary.

If there is still a penumbra to save, the strategy is to restore the blood flow with a recanalisation therapy (i.e. remove the thrombus). There are 4 possible options:

Thrombolysis The patient receives an injection of a product which dissolves the thrombus and thereby restores the blood flow. The product in current is tissue plasminogen activator (tPA) [15]. tPA cleaves the fibrogen which is holding the blood clot together and thus dissolves it. This process can take either some minutes or several hours, depending on the thrombus size, composition and the surface which can be attacked by tPA.

Collateral circulation delivers tPA to the rear end of the thrombus, so it can be attacked from two sides. Even though there is currently no clear proof of the beneficial effect of good collateral circulation on the tPA-induced re-canalisation rate, a good collateral circulation is correlated with a better recovery of the patients [16]. A partial occlusion even allows tPA to attack the thrombus on its whole length, resulting in fast disintegration of the thrombus [17]. The treatment is simple to apply and has a high success rate for small blood clots.

The main risk associated to this treatment is haemorrhage [18], especially symptomatic intracerebral haemorrhage. Fibrogen is also present in vessel

[4]Indeed some people might prefer to rather die than be trapped in a unresponsive wakefulness syndrome, or to forget who their parents are.

walls and weak vessels can be destroyed by tPA. Also tPA will dissolve all blood clots in the body. For example if a patient has a traumatic brain injury because he fell over during the stroke onset, he might have blood clots in the brain which prevent actual haemorrhage. Also women who have their period might experience increased bleeding from tPA.

Furthermore there are thrombi which might not react to tPA because they are not held together by fibrogen. In this case the treatment will fail.

Sequential Double IV Thrombolysis This emerging technique is an extension to thrombolysis. It is developed at the Centre Hospitalier Sud-Francilien (CHSF) and currently an experimental treatment. One hour after thrombolysis with tPA, the re-canalisation success is controlled via MRI. If the thrombus persists, a dose of tenecteplase is administered to speed up the dissolution of the thrombus [19]. This method has seen good results, and is a prime choice for resistant thrombi which are too distal for thrombectomy.

The risk of haemorrhage for this treatment is even higher as for single thrombolysis and has to be considered carefully.

Thrombectomy A recent development, the thrombectomy was a revolution for the field of stroke treatment. It is the mechanical removal of the thrombus. Usually this is done with a catheter which is inserted in the bloodstream and pushed to the position of the occlusion. The approaches of how to extract the thrombus with the catheter can be divided into stent retriever and contact aspiration methods [20].

The re-canalisation rate achieved with thrombectomy is around 80% [21]. But it requires an experienced team of radiologists, who are available in tertiary centers only. Furthermore it is currently limited to large arteries, due to technical problems with small and distal vessels [22]. The outcome of thrombectomy can be improved in combination with thrombolysis [21].

When there is no neuro-interventional team on site, the patient has to be transferred to another hospital. This delays the treatment and gives the infarct more time to grow.

Inserting a catheter in the brain always risks rupturing a vessel. The interaction with the thrombus can also damage the vessel walls. Finally the thrombus can break apart during the procedure and send smaller fragments to more distal locations, causing multiple occlusions.

No Treatment As it might sound surprising, it is the best option for some patients. When the risk of haemorrhage of the three methods above is too high, the amount of tissue destroyed by the haemorrhage might be higher

than the amount of tissue saved by a re-canalisation. This is the sad case of having to accept to leave a part of the brain to die in order to save the rest.

From the options for treatment one must be chosen for each patient. There are official guidelines for the line of action [23]. But they need to be viewed as a rule of thumb and legal coverage for the treating medic. The treatment decision has always to be an individual one.

The optimal treatment in the presence of a penumbra can be found by considering four factors:

- Reactivity of the thrombus to the treatment
- Risk of haemorrhagic transformation
- Time to re-canalisation
- Thrombus location

The reactivity of the thrombus to thrombectomy can be regarded as 100%. The reactivity of the thrombus to thrombolysis depends on the composition and size of the thrombus, the type of occlusion and the amount of collateral circulation.

Thrombi with a high fibrogen content will be dissolved faster by tPA. The thrombus size is important, as dissolving more material takes more time. The bigger the surface on which tPA can work simultaneously is, the faster the thrombus is dissolved.

Haemorrhagic transformations are the most dangerous complications which are associated with the treatment of stroke [24]. The main factors giving rise to an increased haemorrhage risk are old age, infarct volume, and chronic renal failure [25] . Other risk factors are high blood pressure, a high glucose level and major stroke severity [8]. Also the amount of microbleeds are a strong indicator for the haemorrhage risk [26, 27]. In most cases the risk of brain damage by haemorrhage is clearly too low to abstain from recanalisation therapy.

The time to re-canalisation is given by the time it takes to make the treatment decision and to get the the patient to the treatment, plus the time of the treatment. At last the location of the thrombus decides whether thrombectomy is a viable option.

The following set of rules leads to the final treatment decision:

- If the risk of haemorrhage at re-canalisation is small enough, re-canalisation therapy will be attempted.
- If the risk of haemorrhage is too large for thrombolysis, or the thrombus will not react to tPA, therapy will be by thrombectomy.
- If the thrombus is too distal for thrombectomy, therapy will be by thrombolysis.

In the remaining cases the time to re-canalisation will be considered, and the quicker treatment will be chosen. Double thrombolysis is used if the reaction to thrombolysis is present but slow (at 1 hour after the initial tPA administration), and there is no haemorrhage to be expected.

This idealised decision process requires a lot of detailed information about the patients state. In reality most of this information is only partially or not available. The missing parts are replaced by the experience of the treating doctor. This is why the success of stroke treatment is strongly dependent on the diagnostic equipment and the experience of the team at the treating hospital.

1.1.3 Diagnostic Strategies

There are various techniques to help with the diagnosis of the physiology of stroke which is so important for the treatment decision (section 1.1.2). Most of them are imaging methods based either on computer tomography (CT) or magnetic resonance imaging (MRI). But even without machinery some effects of stroke can be observed.

Before any imaging is applied there is a clinical test, where the ability of the patient to move his body parts, to speak, to comprehend, and his reflexes and sensory are tested. The results are combined into a scale called National Institutes of Health Stroke Scale (NIHSS, [28]), which gives a rating of the severity of the stroke. It also gives a prediction for the patient recovery without treatment. For example a NIHSS greater than 16 indicates a very high probability of death [29]. Taking the test results together with an functional atlas of the brain allows to roughly locate the area affected by the stroke. But this is only a rough region, which does not allow to differentiate between dead and still living but function-less tissue. Furthermore there is

no possibility to differ between haemorrhagic and ischaemic stroke without imaging.

Based on CT, there are two methods:

Non Contrast CT (NCCT) CT by itself shows regions of dead tissues (ionic oedema, [30]) in the brain. The thrombus itself can also be seen [31], supposing a slice thickness smaller than the thrombus. This allows estimations of the infarct size and the thrombus size. But the interpretation of NCCT is very difficult and requires a very well trained radiologist.

CT Angiography (CTA) After injection of a contrast agent, which absorbs the x-rays, the arterial tree can be made visible. In a stroke scenario parts of the arterial tree are missing. With sufficient anatomical knowledge these missing arteries can be guessed. This gives away the location of the thrombus and allows an assessment of the penumbra. Note that the penumbra is only a rough guess based on the radiologists and neurologists experience. Also the missing arteries might not be identifiable due to the asymmetry of the arterial tree.

MRI methods allows access to more different physical properties of the examined tissues. This resulted in a series of MRI modalities which are useful for diagnosing stroke:

Diffusion Weighted Imaging (DWI) On the DWI regions with restricted diffusion can be seen (cytotoxic oedema, [30]). This region corresponds to the infarct and part of the penumbra and is also called **lesion**. On Apparent Diffusion Coefficient (ADC) maps, which are calculated from multiple DWIs, the absolute value of the diffusion coefficient can be accessed.

Fluid-Attenuated Inversion Recovery (FLAIR) FLAIR is a very anatomical image. It allows to see the collateral arteries, which is a good way to guess the penumbra [16]. And it serves to distinguish the lesions seen on DWI into chronic and acute damage. This is important if there are lesions from old stroke accidents, or other diseases like cancer.

Time of Flight Angiography (ToF) The arterial tree can also be depicted with ToF. The information gained is identical to CTA. But the image quality is slightly worse, due to technical limits (see section 1.2.4).

Susceptibility-Weighted Angiography (SWAN) Part of the blood flow and the thrombus is visible on the SWAN image. This gives away the thrombus size and location and the type of occlusion.

At last there is one method which can be performed with either CT or MRI:

Perfusion Weighted Imaging (PWI) In PWI the speed with which the tissues are perfused can be measured. Considering that the penumbra and infarct are regions of hypoperfused tissue, these show up on PWI. But as the perfusion is not equal in all the brain there rests some uncertainty.

With all these diagnostic methods, the thrombus location and size and the occlusion type can be determined in theory. Also the infarct itself as well as most of the penumbra and the collateral arteries can be recovered. The only feature which is not available via medical imaging is the composition of the thrombus. This constitutes already a good basis for the decision process outlined in section 1.1.2.

1.2 Diagnosis on MRI

As this work is based on the analysis of magnetic resonance images all other imaging techniques are forgone. This chapter is about the magnetic resonance imaging (MRI) sequences used for diagnosing stroke and how to extract the information from the corresponding images. At first, the basic nuclear magnetic resonance (NMR) experiment will be reviewed. Then the most basic pulse sequences and spatial resolution of the NMR signal will be explained. The last part concerns the basic ideas of the sequences used for stroke analysis and the diagnosis on the resulting images.

1.2.1 The Basics of MRI

MRI is a spatially resolved version of the NMR experiment.

The NMR Experiment

For a detailed explanation of the NMR experiment the book 'Spin Dynamics: Basics of Nuclear Magnetic Resonance' by Levitt [32] can be recommended. It contains a classical as well as a quantum physics derivation of the NMR Basics, an introduction to the required maths and (quantum) physics and a detailed description of the most common pulse sequences and experimental setups. The following summary is based on this book.

Clinical MRI measures only hydrogen atoms. Thus only spin 1/2 particles,

which correspond to the proton of a hydrogen atom, will be considered by now. In the following part, the basic equation of motion of the spin of a spin 1/2 particle in an external magnetic field is outlined. For detailed derivations please consult the above textbook.

The two eigenstates of the spin operator $\hat{I}$ of a single spin 1/2 particle are denoted as:

$$|\alpha\rangle$$
$$|\beta\rangle$$

with the eigenvalues

$$\hat{I}\,|\alpha\rangle = \frac{1}{2}\,|\alpha\rangle$$
$$\hat{I}\,|\beta\rangle = -\frac{1}{2}\,|\beta\rangle$$

Any state $|\psi\rangle$ of the spin can be written as a superposition of the eigenstates:

$$|\psi\rangle = c_\alpha\,|\alpha\rangle + c_\beta\,|\beta\rangle$$

with the complex coefficients c_α,c_α and the normalisation constraint

$$|c_\alpha|^2 + |c_\beta|^2 = 1$$

The eigenstates are usually degenerate, but in the presence of a external magnetic field (B_0) their energy levels split (**Zeeman effect**). Without loss of generality consider the magnetic field to be in direction z. The Hamiltonian in this simple case is given by:

$$\hat{H}_0 = \omega_0 \hat{I}_z$$

which depends only on the spin operator in z-direction. The energy of the eigenstates is

$$\hat{H}_0 |\alpha\rangle = \frac{1}{2}\omega_0 |\alpha\rangle$$
$$\hat{H}_0 |\beta\rangle = -\frac{1}{2}\omega_0 |\beta\rangle$$

with the Larmor frequency $\omega_0 = -\gamma B_0$. γ is the gyromagnetic ratio. The difference of the energy levels is therefore ω_0 and corresponds to the energy needed or gained by changing the state $|\beta\rangle$ to $|\alpha\rangle$.

The equation of motion of a spin in state $|\psi\rangle$ is given by the Schrödinger equation:

$$\frac{\mathrm{d}}{\mathrm{d}t} |\psi\rangle = -i\hat{H}_0 |\psi\rangle$$

which has the solution

$$|\psi\rangle (t) = \exp\left(-i\omega_0 t \hat{I}_z\right) |\psi\rangle (t_0) = \hat{R}_z(\omega_0 t) |\psi\rangle (t_0)$$

This corresponds to a rotation in the x-y plane about the angle $\omega_0 t$, given by the rotation operator $\hat{R}_z(\phi)$. So a spin will precess around the z axis with frequency ω_0, once its value in z direction has been measured.

Now consider the interaction with an external radio frequency (RF) pulse of phase ϕ_p, oscillating at frequency $\omega_{RF} = \omega_0$. The time dependent angle Φ_p is given by $\Phi_p(t) = \omega_{RF} t + \phi_p$. θ_{RF} denotes the angle of the RF pulse to the B_0 field and B_{RF} is the maximum field strength of the oscillating RF field. The Hamiltonian for this case is given by

$$\hat{H} = \hat{H}_0 + \hat{H}_{RF}$$
$$\hat{H}_{RF} = -\frac{1}{2}\gamma B_{RF} \sin(\theta_{RF}) \hat{R}_z(\Phi_p) \hat{I}_x \hat{R}_z(-\Phi_p)$$

and the corresponding Schrödinger equation by

$$\frac{\mathrm{d}}{\mathrm{d}t} |\psi\rangle = -i\hat{H} |\psi\rangle$$

When the pulse starts at t_1 and ends at t_2 with duration τ, the solution for the change of $|\psi\rangle$ during the pulse is

$$|\psi\rangle\,(t_2) = \hat{R}_z(\Phi_p(t_2))\hat{R}_z(\phi_p)\hat{R}_x(\beta_p)\hat{R}_z(-\phi_p)\hat{R}_z(-\Phi_p(t_1))\,|\psi\rangle\,(t_1)$$

with $\beta_p = \omega_{nut}\tau = |\frac{1}{2}\gamma B_{RF}\sin(\theta_{RF})|\tau$. The spin performs a nutation movement around the z-axis with the nutation frequency ω_{nut} during the RF pulse. So the effect of an RF pulse is to increase the angle between the spin and the z-axis. The longer the RF pulse, the larger is the increase.

Note that the effect of the RF pulse depends on the angle of the pulse to the B_0 field.

In the case of $\omega_{RF} \neq \omega_0$, the effect is the same, but the strength of the effect (i.e. the change in angle of the spin) decreases drastically with increasing difference $|\omega_{RF} - \omega_0|$.

In a real material, the magnetic field experienced by the spins in a nucleus is modified by the other particles in the nuclei, the electron hull of the atom and nearby atoms. These atoms are usually those from the molecule or crystal lattice, the nucleus is embedded in. Therefore the Larmor frequency changes to

$$\omega_0 = -\gamma B_0(1+\delta)$$

with the chemical shift δ. The effect of the chemical shift can be used to determine the structure of molecules.

As an NMR experiment measures a sample of finite size it is mandatory to look at the situation of multiple spins in a magnetic field. The spins in the sample are considered as non interacting, for simplicity. The state of the spin ensemble is described by the spin density operator

$$\hat{\rho} = \begin{pmatrix} \overline{c_\alpha c_\alpha^*} & \overline{c_\alpha c_\beta^*} \\ \overline{c_\beta c_\alpha^*} & \overline{c_\beta c_\beta^*} \end{pmatrix} = \begin{pmatrix} \rho_\alpha & \rho_+ \\ \rho_- & \rho_\beta \end{pmatrix}$$

where the overhead bars stand for the average over the whole ensemble and c_α^* for the complex conjugate of c_α. This is a way to specify the states $|\psi\rangle$

of all spins in the ensemble by averaging over the respective c_α and c_β. The values of the spin density operator allow statements about the macroscopic behaviour of the spin ensemble.

The ρ_+ and ρ_- are called coherences. They correspond to a net polarisation perpendicular to B_0. Here the phase of the complex number ρ_- indicates the angle of the polarisation. In thermal equilibrium $\rho_+ = \rho_- = 0$, which means no polarisation in x or y direction. An existing coherence means that the precession of the individual spins is, at least somewhat, aligned. In this case it is said the spins are in phase.

The ρ_α and ρ_β correspond to the population of the eigenstates $|\alpha\rangle$ and $|\beta\rangle$ of $\hat{I}_z$. Note that most spins are in superposition states $|\psi\rangle = c_\alpha |\alpha\rangle + c_\beta |\beta\rangle$ and the population corresponds to the average values of $|c_\alpha|^2$ and $|c_\beta|^2$. If the two populations are not equal this means there is an average polarisation of the spins in direction of the magnetic field. It turns out that the populations follow the Boltzmann statistics in thermal equilibrium. The populations in thermal equilibrium can therefore be approximated by

$$\rho_\alpha^{equilibrium} = \frac{1}{2} + \frac{1}{4}\mathcal{B}$$
$$\rho_\beta^{equilibrium} = \frac{1}{2} - \frac{1}{4}\mathcal{B}$$

with the Boltzmann factor

$$\mathcal{B} = \frac{\hbar\gamma B_0}{k_b T}$$

Here $\hbar$ is the reduced Planck constant, k_b the Boltzmann constant and T the temperature. So at thermal equilibrium there is indeed a net polarisation of the spins along the magnetic field.

Globally speaking, the spin density operator represents the macroscopic magnetisation of the sample arising from the ensemble of the individual spins. The part in direction of B_0 is given by the populations and the part perpendicular is given by ρ_-.

Under the influence of a RF pulse, the macroscopic magnetisation does very much the same as the individual spin as shown above. The formal presentation of the equation of motion with the spin density operator is omitted as it

is not necessary to understand the concept. The interested reader is referred to chapter 11 of [32].

The angle of the magnetisation to the magnetic field can be modified at will with a RF pulse of correct length at frequency ω_0. Notably the angle can not only be increased but also be decreased. Note that a change of the angle implies an exchange between the diagonal and anti-diagonal elements of $\hat{\rho}$. After the RF pulse the magnetisation vector precesses around the z-axis.

There are some important RF pulses which have been given names corresponding to their action on the magnetisation vector.

The $(\pi/2)_x$ pulse turns the magnetisation vector by 90° around the x-axis. In the case starting in thermal equilibrium this turns the magnetisation from the z-axis into the $-y$ axis. This equalises the populations and transfers their entire difference to the coherences.

The π_x pulse turns the magnetisation vector by 180° around the x-axis. In the case starting in thermal equilibrium, the pulse inverts the magnetisation vector, so it points in $-z$ direction.

More general, an $(\beta_p)_x$ RF pulse turns the magnetisation vector around x-axis by the angle $\beta_p = \omega_{nut}\tau_p$ where τ_p is the length of the pulse. The subscript $()_x$ denotes a rotation around the x-axis. This can be the y-axis as well. If the subscript is not mentioned this means the axis of the rotation is not important but it always lies in the x-y plane.

As any physical system, a spin ensemble, which has been disturbed from its equilibrium state by a RF pulse, converges back to equilibrium. Both population and coherence decay exponentially.

The decay of coherence is due to random small local fluctuations in the magnetic field. This causes small variations in the precession frequency ω_0. Remember that ω_0 depends on the magnetic field experienced by the spin. The different precession speeds destroys the alignment of the precessing spins, they dephase. This decay is called **transversal** or **spin-spin relaxation** and decays with time constant T_2. No energy is transferred but entropy increases. Therefore it is irreversible.

The decay of population, known as **longitudinal** or **spin-lattice relaxation**, decays with time constant T_1. The decay of the magnetisation in z direction requires moving portions from the higher energetic state $|\beta\rangle$ to $|\alpha\rangle$. The released energy is exchanged with the surroundings via multiple mechanisms and transferred to vibration and rotation of molecules. For example with the lattice in solids, hence the name spin-lattice relaxation. Of course

this energy exchange is not reversible. For time t after the end of the RF pulse the decay equations are

$$\begin{aligned}
\rho_\alpha(t) &= (\rho_\alpha(0) - \rho_\alpha^{equilibrium})e^{-\frac{t}{T_1}} + \rho_\alpha^{equilibrium} \\
\rho_\beta(t) &= (\rho_\beta(0) - \rho_\beta^{equilibrium})e^{-\frac{t}{T_1}} + \rho_\beta^{equilibrium} \\
\rho_+(t) &= \rho_+(0)e^{(-i(\omega_0-\omega_{RF})-\frac{1}{T_2})t-i\Phi(t)} \\
\rho_-(t) &= \rho_-(0)e^{(+i(\omega_0-\omega_{RF})-\frac{1}{T_2})t+i\Phi(t)}
\end{aligned}$$

where $\Phi(t) = \Phi_p(t + (t_2 - t_1))$.

In measurements of real samples the observed T_2 times are much shorter than the theoretical expectation. They are modified by static variations to the magnetic field cause by imperfections in the magnet which generates the B_0 field and magnetic substances in the sample, which also modify the local magnetic field. A spin experiences a local magnetic field $B_l = B_0 + \Delta B$. The modified lateral relaxation time T_2^* is defined as

$$\frac{1}{T_2^*} = \frac{1}{T_2} + |\gamma \Delta B|$$

In contrast to the dephasing caused by random fluctuations, the dephasing by static differences in the magnetic field is reversible by reversing the direction of precession. This is used to design experiments which measure T_2 or T_2^* explicitly.

As any moving magnet, the precessing magnetic moment of the spin ensemble creates a RF signal, called the NMR signal. The NMR signal s is proportional to the coherence ρ_-:

$$s \propto 2i\rho_-$$

Pulse Sequences

Armed with the knowledge about the magnetisation vector, how to manipulate it wit an RF pulse and how the resulting signal looks like, it is possible to

design pulse sequences in order to measure specific effects. This part explains the most basic pulse sequences. The pulse sequences are always assumed to start in a state of thermal equilibrium.

Inversion Recovery The inversion recovery sequence is designed to measure the T_1 relaxation time. Remember that the NMR signal is proportional to the coherence ρ_-. Therefore T_1 cannot be measured directly.

The trick is to invert the populations with a π pulse. The magnetisation in z-direction starts to decay with T_1. After a time τ the magnetisation is turned into the x-y plane by a $\pi/2$ pulse. The population difference has been entirely converted to coherence. The amplitude of the now measurable signal corresponds to the magnetisation in z-direction at time τ but decays with T_2^*.

Repeating this experiment with different delays τ allows to track the decay of the magnetisation in z direction. Of course one needs to wait for the thermal equilibrium to be reestablished between two experiments. From the resulting curve T_1 can be calculated.

Spin Echo In order to measure T_2 the effects of the static imperfections of the magnetic field need to be negated. The reversible nature of the static imperfections makes this possible.

The pulse sequence starts with a $(\pi/2)_x$ pulse, turning the magnetisation into $-y$ axis. The magnetisation now decays with T_2^*. After a time delay $\tau/2$ a π_y pulse is applied. This has the effect of flipping the spins about 180° around the y-axis. The amplitude of the magnetisation remains untouched, but the direction of the precession is reversed. The spins still precess with the same frequency as before given by the local magnetic field. After waiting another period of $\tau/2$ the spins rephase, and for one instant the effects of the magnetic field imperfections are negated. The difference of the amplitude at this moment to the starting amplitude is solely due to the lateral relaxation.

Repeating the experiment for different values of τ enables the calculation of T_2. As usual the thermal equilibrium has to be reached between two experiments. If the NMR signal is measured during the experiment the signal decays at first with T_2^* and then rise again after the π_y pulse due to the rephasing spins. Therefore the name spin echo. After the spins have rephased the signal decays again with T_2^*.

Gradient Echo The idea of the gradient echo is the same as for the spin echo. The precession direction of the spins is reversed and the rephasing allows to measure the amplitude due to T_2 alone.

As the precession frequency $\omega_0 = -\gamma B_0$ is proportional to the magnetic field, inverting the field inverts the direction of precession. The gradient echo starts with a $\pi/2$ pulse. After the time period $\tau/2$ the magnetic field is inverted and after another period of $\tau/2$ the signal is measured.

Once again T_2 can be calculated from the amplitudes of repeated experiments with different τ. As before the thermal equilibrium must be ensured between two consecutive experiments.

Achieving Spatial Resolution

If a NMR experiment is done, there is only one resulting signal for the whole sample. In other words, the spin density operator represents the spins of the whole sample. As such this is not very useful for medical imaging. The invention of the spatial encoding of the NMR signal by Lauterbur and Mansfield [33, 34] was awarded with the Nobel price in 2003. They developed a technique where the spin density operator can be made to represent only a portion of the sample. As the space for real applications has the dimension 3 the encoding for the location of the measurement takes 3 steps.

Slice Selection Remember that the Larmor frequency ω_0 depends on the field strength of the applied magnetic field. If an additional magnetic gradient field $B_g(z)$ is applied in z direction, the resulting Larmor frequencies $\omega_0 = -\gamma(B_0 + B_g(z))$ now depend on the z position.

The action of an RF pulse with frequency ω_{RF} on a spin with Larmor frequency ω_0 decreases rapidly with the difference of the frequencies $\Delta = |\omega_0 - \omega_{RF}|$. So an RF pulse in the presence of a gradient field B_g acts only on those spins whose ω_0 matches ω_{RF} and those with a very small Δ. In geometrical terms this means only a slice of the sample, within a certain range of z is excited by the RF pulse.

The thickness of the slice depends of the strength of the gradient field B_g. As in reality no RF pulses with an infinitely sharp spectrum can be created the spectral width of the RF pulse increases the width of the slice also.

The gradient for the slice selection is on during the exciting pulse sequence. It is off during the measurement of the signal. This ensures, that only spins of one slice are excited and that the measurement of the signal is not disturbed by the gradient.

Frequency Encoding The selection of the slice takes place during the pulse sequence. The same idea can be used to partition the slice into rows, if

applied during the measurement.

During the measurement phase, an gradient field $B_x(x)$ is applied along the x-axis. This affects the Larmor frequencies $\omega_0 = -\gamma(B_0 + B_x(x))$ to be dependent on their position on the x-axis. Consequently the measured NMR signal is now a mixture of all the Larmor frequencies found along the x-axis.

The amplitude corresponding to a specific row can then be found by analysing the Fourier transform of the NMR signal. The maximum resolution achievable with frequency encoding and slice selection depends on the strength of the gradient fields and the bandwidth of the RF pulse

Phase Encoding The last direction cannot be recovered via the frequency information. Here the phase of the signal is used.

Applying a gradient field $B_y(y) = G_y y$ in y-direction once again causes the spins to precess with different Larmor frequencies $\omega_y = -\gamma G_y y + \omega_0$. Due to the different precession speeds the spins dephase with respect to the y-axis but stay in phase along the x-axis. The longer this gradient is active, the higher the difference in phase becomes. The phase difference accumulated in time τ is equal to $-\gamma G_y y \tau$.

The phase of the spins cannot be measured directly. Recall the NMR signal of an ensemble of spins:

$$s \propto 2i\rho_- = 2i\rho_-(0)e^{(+i(\omega_0-\omega_{RF})-\frac{1}{T_2})t+i\Phi(t)}$$

To facilitate the notation let the RF pulse be of frequency $\omega_{RF} = \omega_0$, let the initial phase factor $\Phi(0) = 0$ and assume that the signal is measured at $t = 0$:

$$s(0) \propto 2i\rho_- = 2i\rho_-(0)$$

Now consider the effect of the gradient field $B_y(y)$. It adds a phase of $-\gamma G_y y \tau$ to the signal of the spin ensemble at position y:

$$s(0) \propto 2i\rho_- = 2i\rho_-(0)e^{-i\gamma G_y y \tau}$$

The signal of all spin ensembles along the y-axis is given by

$$s(0) \propto 2i\rho_- = 2i\rho_-(0) \int_y e^{-i\gamma G_y y\tau} dy$$

The important point here is that the term

$$\int_y e^{-i\gamma G_y y\tau} dy$$

can be interpreted as a superposition of waves with frequencies $\gamma G_y y$ at time τ. Measuring the signal at $t = 0$ for different times τ allows to apply the Fourier transform to obtain the phase shifts $-\gamma G_y y\tau$. The minimum number N of different τ_i which allows to differentiate N points along the y-axis is N. This stems from the Nyquist theorem for the sampling rate.

For phase encoding the maximum resolution is also limited by the strength of the gradient fields and the bandwidth of the RF pulse.

From Encoding to Images

After seeing how the three spatial dimensions are encoded, they can be combined into one measurement procedure:

- Select a slice with coordinate z
  - Apply the gradient field B_g
  - Apply the exciting pulse sequence
  - Shut down the gradient field B_g
  - Only spins in slice z are excited and will emit a signal

- Prepare phase shift for τ_i
  - Apply the gradient field B_y for the time τ_i
  - Shut down gradient field B_y

- Prepare rows along x-coordinate
  - Wait for the echo from the exciting pulse
  - In that moment apply the gradient field B_x
  - And measure the signal s(z,τ_i,x)

The resolution in y-direction is determined by the number M of τ_i. In x-direction the resolution is determined by binning the signal into N bins. For each slice the measurement procedure has to be repeated M times. Each slice's results are collected in an $M \times N$ matrix. This matrix is called k-space matrix.

The MRI image for one slice is obtained by the 2D Fourier transform of the corresponding k-space matrix. Assuming L slices the effort for this procedure is $L * M$ measurements. Images obtained by this procedure are called 2D images, as the volume is covered by a sequence of 2D slices.

One drawback of this method is the time the spins need to return to equilibrium. To get acceptable measurement times for clinical use there are gaps left between the slices. This speeds up the measurement by omitting parts of the volume. It also avoids crosstalk with the spins from a previously selected slice and therefore the consecutive slices can be measured faster.

3D MRI In contrast to the slice and row selection by gradients the phase encoding technique is not limited to a specific number of dimensions. All three dimensions can be encoded by phase. The resolution is then limited by the number of measurements, up to the technical limits. But the effort increases to $L * M * N$ measurements.

MR images from triple phase encoding are called 3D images, as they measure the whole volume in one measurement and it uses a 3D Fourier transform of the k-space matrix. Which has now the size $L \times M \times N$.

3D images usually have a higher image quality as the noise decreases with the amount of measurements. They also do not suffer from cross-talk artefacts and no gaps are left in the measured volume. But they take longer to acquire as 2D images of the same size. Also motion of the scanned object, or patient in the clinical case, destroys the whole volume. In 2D images motion renders only some slices useless.

Further Considerations In order to use any NMR pulse sequence in an MRI experiment special care is needed. The gradients used for the spatial localisation interfere with the gradients of the pulse sequence. Combining the two

requires advanced calculations. These need to take into account the actual layout of the machine which generates the magnetic fields. It is out of the scope of this work to talk about the construction details of an MRI machine.

Usual field strengths of the B_0 field are 1.5 and 3 Tesla for routine clinical use. Research scanners may have field strengths up to 7 Tesla. These high field strengths can currently only be acquired with supra-conducting electromagnets, which in turn need liquid helium for cooling. Of course these high fields disturb nearby electrical equipment, making an electromagnetic shielding of the room necessary. This is done with massive copper plates. An MRI machine is thus a very expensive machine.

The edge lengths of the voxels range from 0.5mm to 4mm in clinical scanners. They could be smaller in theory but are limited by constraints to the scan time. In a stroke examination this is the urgent need for a treatment. Most other examinations could afford longer scan times. But a hospital needs to make the scanner available for the most exams possible in order to reduce the costs of a scan per patient. Therefore the resolution is reduced to the minimum point where the medical doctors can still see the deceases they are looking for. This is the reason why clinical MRIs are often of low quality.

1.2.2 DWI and ADC

Diffusion weighted imaging was developed in France in 1984 by Denis Le Bihan [35]. Some years later it was discovered that ischaemic lesions in cat brains were visible on DWI. After some trials it was quickly adapted to the medical routine for stroke examinations.

Basic Idea Only the basic idea of DWI is described here. An elaboration of diffusion on cellular level and the mathematics leading to the different pulse sequences in use can be found in the review by Valerij Kiselev [36].

Spins which are moving during a spin echo experiment accumulate an additional phase. This is due to field inhomogeneities they traverse which in turn constantly changes their Larmor frequency by small amounts. Consequently their spin echo signal is reduced, *i.e.* they appear darker on the MRI. The greatest phase shifts are acquired by spins moving in direction of the gradient field, if a gradient echo sequence is used. This effect of the diffusion along the gradient field can be increased with higher gradient fields to the point where this effect dominates the image. In order to get rid of the directional dependency the measurement is repeated with gradients in different directions and S is averaged over at least 6 measurements. The resulting image

is called diffusion weighted image (DWI) and is bright for slow diffusion and dark for quick diffusion.

At the cellular level, diffusion is a very complex process which depends on the local micro-structure. Most importantly it is not isotropic. The diffusion at one point is given by the diffusion tensor. The interaction of the diffusion tensor and the NMR signal is quite complicated. For simplicity all the interactions are summed up in one variable, the so called b-value. This is largely dominated by the applied gradient field and thus b can be identified with the gradient field for practical purposes. The signal amplitude can then be approximated as:

$$S = e^{-bD} \tag{1.1}$$

where D is the apparent diffusion coefficient. It is a very crude model, which ignores all cellular structure. b is dependent not only on the gradient fields but also on time. Also D is a snapshot of the diffusion and subject to random movement of the molecules. Therefore two measurements of D will result in different values.

The apparent diffusion coefficient (ADC) can be recovered by looking at the logarithm of the signal:

$$\ln(S) = -bD$$

This is now a linear equation of b and D. By measuring the signal at different values of b, D can be calculated with simple regression. The ADC is a quantitative measurement and thus studies of the numerical values are possible.

The gold standard pulse sequence is echo planar imaging (EPI) and is used for almost all clinical DWI [37]. It takes around 100ms and thus motion artefacts are mostly eliminated.

Interpretation of DWI In consequence of equation 1.1, voxels with lower diffusion coefficient have a higher value on the final image. As stated in section 1.1.1, neural tissue suffering from cytotoxic oedema has a lower diffusion coefficient. Suppose the normal value of a tissue type is known then the cytotoxic oedema can be seen as a region with increased value. Unfortunately, as stated above, repeated DWI measurements give different numbers.

Thus the normal value of the tissues cannot be known. Nevertheless the variation in subsequent measurements is smaller than the increase in value of cytotoxic oedema. So the regions with cytotoxic oedema appear whitish to bright compared to the surrounding normal tissue. The part of the oedema visible on DWI is also called **lesion**.

Another difficulty for the interpretation of stroke DWI is the bad resolution. DWI in the CHSF stroke data-set (section 4.1) has a slice resolution of 256×256 with 30 to 36 slices, depending on the brain height, and a slice thickness of 3.6mm. In a voxel of 3.6mm height there are thousands of neurons. So the signal is the average over all the neurons in a voxel. If only a part of the neurons of a voxel are in the state of a cytotoxic oedema, the voxel's value will be only slightly increased. So the oedema on DWI will not have a sharp border, but a gradual transition from normal to oedema.

So a lesion on DWI is an area brighter than the surrounding tissues of same type in the same slice. Note that there is a small difference for the average normal value from white to grey matter and the lentiform nucleus has a lower base value. A developing oedema in the lentiform nucelus might look like normal white matter but is already in a cytotoxic state. An example is given in figure 1.2.

As the DWI is inherently an T_2 image, everything which is bright on T_2, will also be slightly brighter on DWI. This T_2 shine through can be eliminated by calculating a so called exponential image, by dividing the DWI by an "DWI" taken with $b = 0$. The natural logarithm of the exponential image is close to the ADC.

Lesions which do not belong to an ischaemic lesion do not show a drop in ADC as their intensity on DWI is due to T2 shine through. Old ischaemic lesions, which continue to the state of ionic oedema, start to appear on the FLAIR image and can thus be told apart from recent lesions (compare figure 1.4).

Artefacts on DWI In contrast to the other MRI modalities artefacts severely disturb the diagnosis on DWI [37]. Therefore a short summary of the most common ones is included here.

Luckily motion artefacts are not much of a problem, as they can be avoided by using modern pulse sequences. More problematic are variations of the magnetic field. The interface between bone and brain tends to create field changes which change the Larmor frequency that much that it changes to a frequency which corresponds to a different row or column in the spatial encoding (see section 1.2.1). Thus the pixels at the interface are displaced

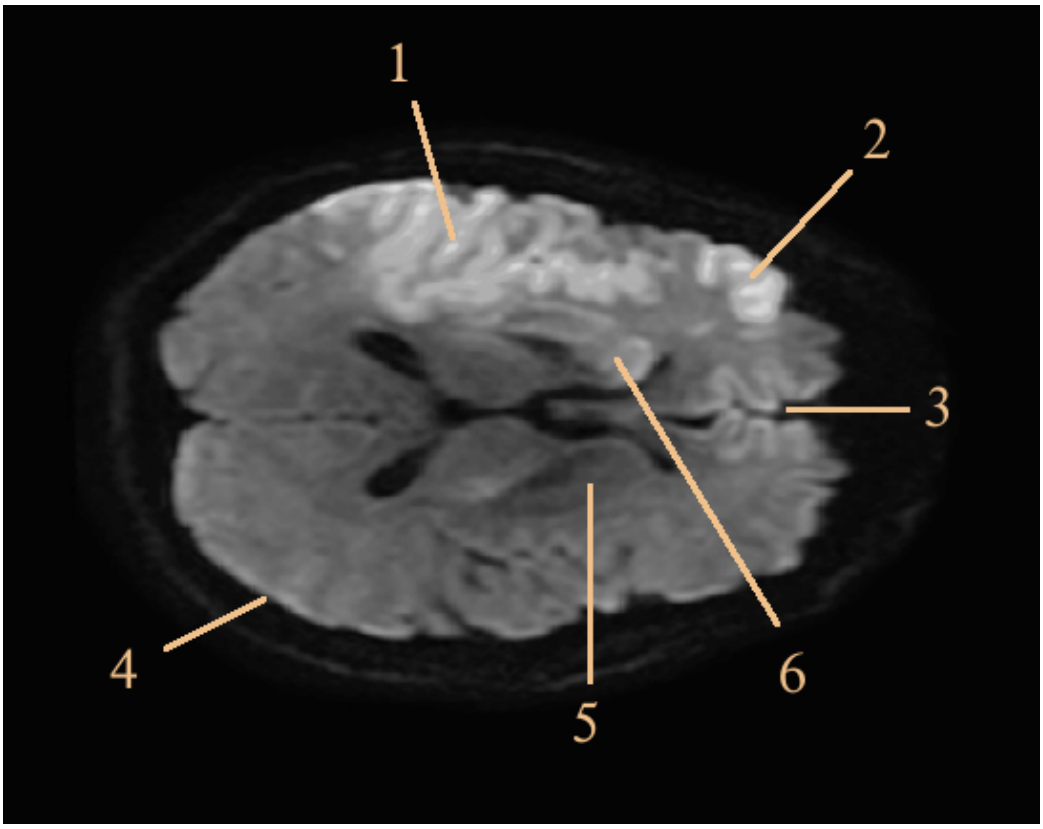


Figure 1.2 DWI slice with a stroke lesion. On this slice the lesion is seen as 2 disconnected regions (1 and 2). Borders of the brain may also have increased intensity (3 and 4) due to partial volume effects and eddy currents. The lentiform nucleus (5) has a lower base value than the surrounding tissue. Therefore the infarcted lentiform nucleus (6) has an intensity close to the normal tissue and is not as bright as other parts of the lesion (1 and 2).

up to 8 pixels [37]. Additionally the amplitude is modified by the magnitude of the shift of the magnetic field making them appear very bright (see figure 1.3).

On the same principle any magnetic field causing a shift larger than the bandwidth of a voxel displaces the voxels in the phase encoding direction. Causes are magnetic fields from tissue interfaces (a change in susceptibility), chemical shift, and eddy currents [38]. They all displace the voxels usually by 1 or 2 in the phase encoding direction. Thus they may accumulate the intensity of several voxels into one, while leaving other voxels empty.

Brain borders are also influenced by partial volume effects from the cardiac motion. Each heartbeat causes the brain to move a little bit. Thus the ratio

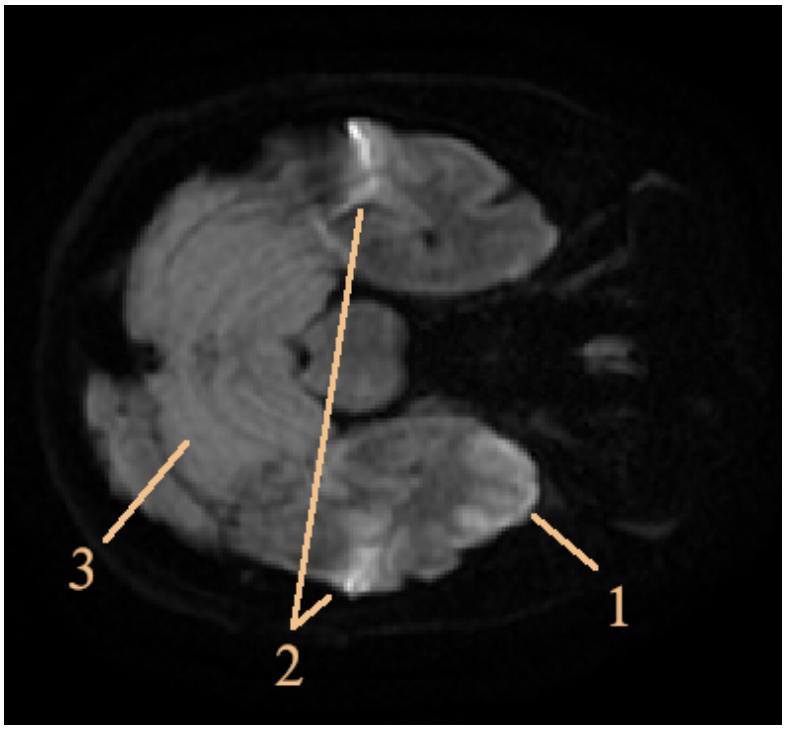
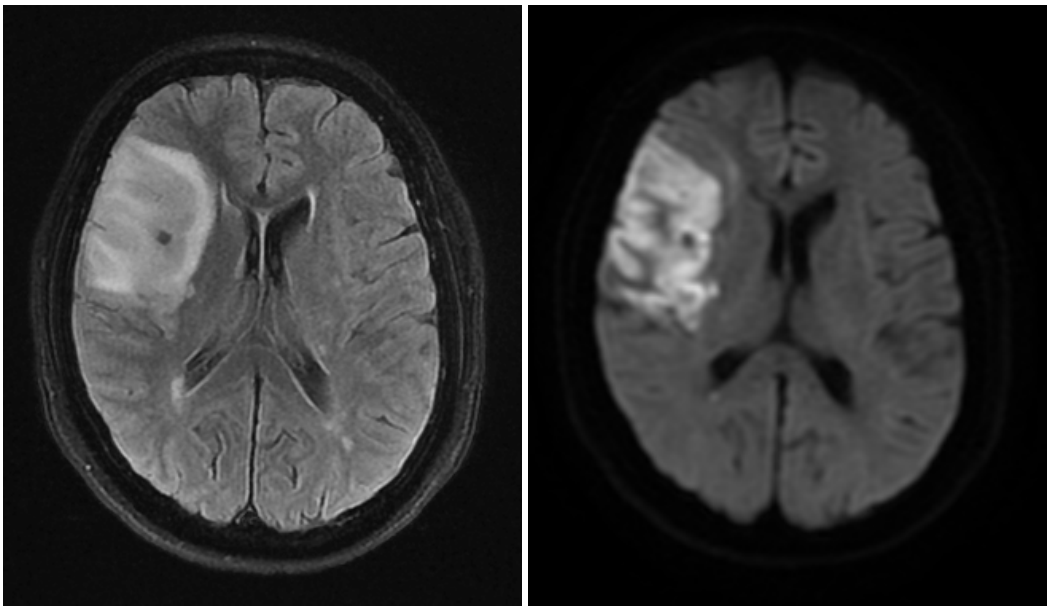


Figure 1.3 Bone brain interface artefacts (2) appear very bright and can be easily confused with nearby lesions (1). Also the cerebellum (3) has a higher intensity in the same range as developing lesions

between brain and cerebrospinal fluid (CSF) in a border voxel pulsates. As the signal intensities are very different for these two this can lead to strong changes between subsequent measurements. Thus these voxels have a high variance which falsifies the calculation of the ADC for these voxels.

Another artefact is the so called T_2 shine-through. It arises from the fact that DWI is basically a T_2 measurement which has been tricked into being sensitive for molecular motion by using very high gradient fields. But the original T_2 contrast still adds to the image. Thus lesions like tumours, which are bright on T_2 weighted images, are also bright on DWI and might be mistaken for ischaemic lesions. Fortunately these do not affect the ADC.

Last but not least are the eddy currents. The change in the magnetic flux from turning on and off the gradient fields induces currents in all metallic parts of the MRI machine. This can result in persisting magnetisation of these parts, which then overlaps with the field in the scanner. Thus the magnetic field differs from the calculated one and the image is distorted. Usually this is seen as shearing, shifts and scaling along a single axis of the



a) b)

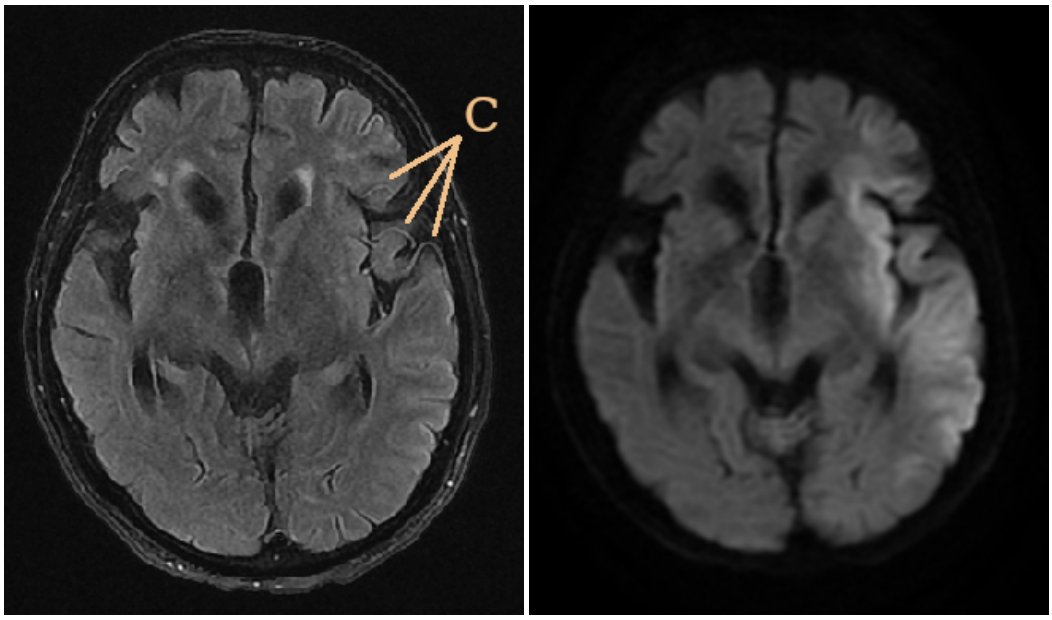
Figure 1.4 A sub-acute stroke lesion becomes visible on FLAIR a) and is very bright with a well defined border on DWI b). In this state it is too late for any treatment. Also the different resolution of FLAIR and DWI becomes apparent in this direct comparison.

resulting image. These can mostly be compensated by shielding the coils and calibrating the machine [37]. But they may still create problems at interfaces between brain and CSF [38].

1.2.3 FLAIR

Fluid attenuated inversion recovery (FLAIR) is another technique from the early age of MRI, published first in 1992 [39]. It consists of a modified T_2 where the signal from the CSF is suppressed.

Basic Idea The T_1 relaxation time of CSF is way longer than those of the brain tissues. This can be used to effectively null the signal of CSF. After an inversion (180°) pulse the magnetisation is inverted and starts to decay with T_1 as described in section 1.2.1. So the magnetisation is anti-parallel to the field B_0. There is also a moment when the magnetisation becomes zero, after a time TI, before it grows in direction of the field. TI is tissue specific, as is the relaxation time T_1.



a) b)

Figure 1.5 On FLAIR a) the collateral arteries (C) are partially visible as fine white lines. They indicate the regions of hypoperfused tissue where in this patient a lesions is already visible on DWI b). It is clear that the actual size of the lesion is largely different from the visible collateral arteries and that they are just an indicator for the lesion.

Notably at the TI_{CSF} of the CSF the brain tissues have already regained much of their equilibrium magnetisation. Thus at TI_{CSF} a spin echo pulse sequence is started. The spins of the CSF have no magnetisation at this moment and thus cannot be flipped while the spins of the brain tissue are flipped. The result is a T_2 spin echo without any signal from the CSF.

With this technique any tissue with known TI can be suppressed. At the cost of tissue with similar TI having a decreased signal. Fat is such a candidate which often obscures information in MRI and thus is often subject to suppression. FLAIR corresponds explicitly to the suppression of the free water in the CSF.

Interpretation FLAIR images have the advantage that partial volume effects in borders with the CSF are absent. So the T_2 values close to brain-CSF borders can be investigated.

All types of lesions which are characterised by a changes T_2 value can be seen on FLAIR. In a stroke scenario it helps to distinguish cytotoxic oedema from

the infarct core. Thus if a lesion is seen on DWI, but not on FLAIR, it might be reversible. If a lesion is seen on DWI and FLAIR and has a below normal ADC value it is part of the infarct core and already dead tissue (figure 1.4). Lesions seen on FLAIR which show no decrease in ADC correspond to other diseases. For example leukoaraiosis is often encountered in connection with stroke [40].

Another useful effect is that the collateral arteries show up on FLAIR [41]. Although the usual low resolution of FLAIR limits the visible amount, especially if they are (partially) in the gaps between the slices. It still allows to estimate the area in which collateral circulation takes place (see figure 1.5.

1.2.4 ToF

Time of flight angiography is in its basic form stems from 1988 [42]. It allows to get a clear signal from flowing blood without the need for a contrast agent. This allows directly to create a map of the arteries.

Basic Idea If a spin is subjected to a series of RF pulses with a short repetition time TR, it cannot reach its equilibrium state between the RF pulses. The magnitude of the magnetisation decreases, until it reaches a steady state magnetisation M_{ss}:

$$M_{ss} = M_0 \frac{1 - e^{-\mathrm{TR}/T_1}}{\cos(\beta_p) e^{-\mathrm{TR}/T_1}}$$

M_0 is the magnetisation in thermal equilibrium and β_p the flip angle of the RF pulse, as described earlier. The series of RF pulses is called the saturation pulse. A T_1 measurement of the saturated spin yields a lower intensity

Now an entire slice is saturated and then T_1 is measured after a short delay. Material flowing into the slice, blood in a clinical application, was not subjected to the saturation pulse. It shows the usual T_1 intensity. Note that flows inside the slice are not visible as these are subject to the saturation pulse.

ToF can also be done in a 3D version with enhanced resolution. Here an entire slab (a 3D volume) is saturated, i.e. the RF pulse contains all frequencies corresponding to the slab. Then again the T_1 is measured. As the slab is bigger than one slice small vessels with slow flow are likely not to receive blood which was outside the slab during the start of the saturation pulse.

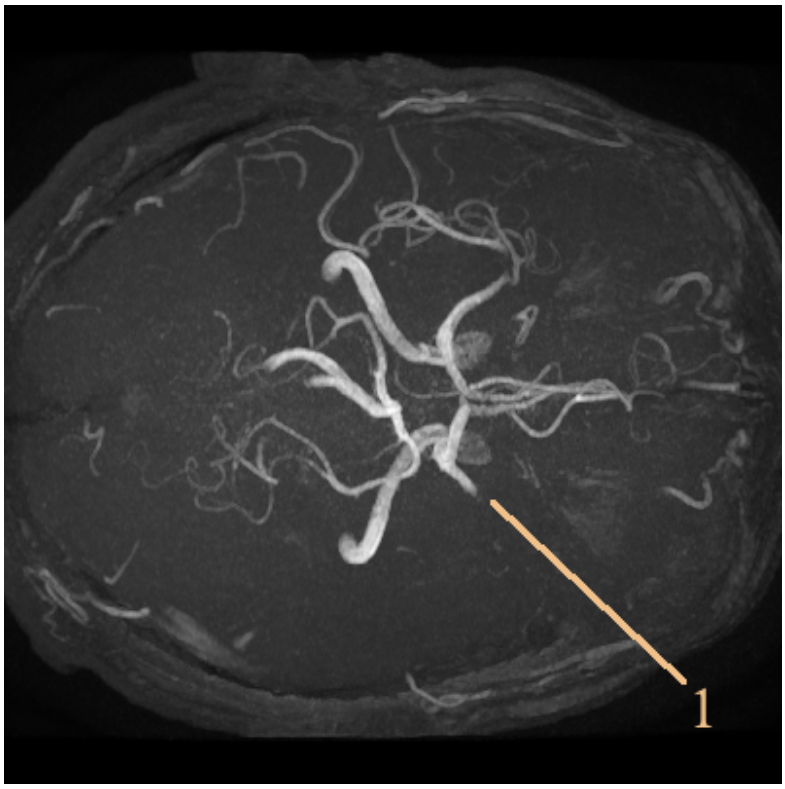


Figure 1.6 Maximum intensity projection of a patient with middle carotid artery (MCA) occlusion. The artery stops suddenly at the point of occlusion (1).

Another effect is that blood which entered the slab during the saturation pulse is not fully saturated and shows only a reduced signal which is still higher than the signal from the saturated tissue. So 3D ToF shows a gradient in the blood vessels depending on the point during the saturation pulse when the blood entered. Imagine the blood flows from below into the slab, and a human head is measured. The blood will be brightest on the entry-points, the carotid arteries. It will then loose intensity, tracing the blood flow with time until the signal drops to invisibility in the smaller branches. Thus human heads are usually measured in two overlapping slabs, to increase the signal of the upper part of the brain [43].

Interpretation Looking at the single slices of a ToF has usually little value. A very skilled radiologist can trace the vessels slice by slice and reconstruct the vasculature in his head. For everybody else there are maximum intensity projections (MIP).

For a MIP the scanned 3D volume is projected to a plane. The intensity of a

point on the plane is determined by the brightest point which was projected onto it from the 3D volume. The projection plane can be rotated freely around the 3D volume, creating the impression of looking at a 3D model of the arterial tree.

With the help of this visualisation tool the arterial tree can be scanned for missing branches or stenosis. A partial occlusion will be visible as a stenosis with a branch of reduced intensity behind it. A complete occlusion corresponds to a missing branch where an artery suddenly stops (figure 1.6). This can be difficult to spot if the thrombus sits just behind a bifurcation.

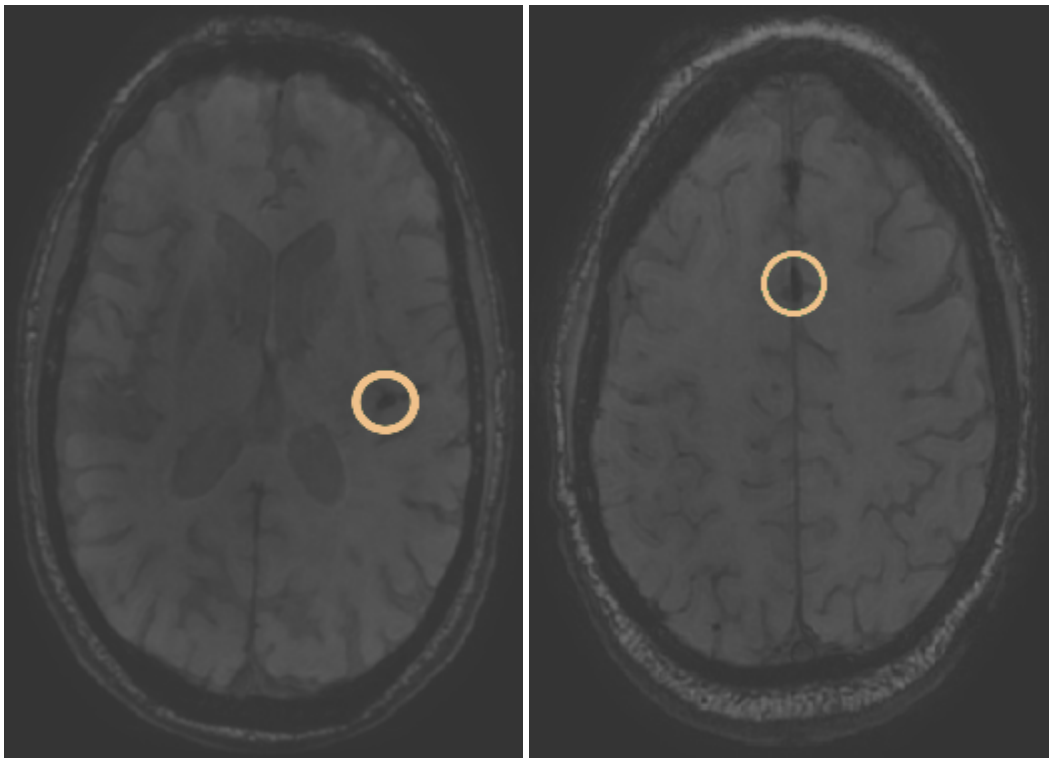
The hypoperfused area can also be estimated by comparing the amount of arteries on both sides of the hemisphere. But this is a very rough guess.

1.2.5 SWAN

Susceptibility weighted angiography is the newest amongst the techniques used for diagnosis in stroke. It is a susceptibility weighted imaging sequence tied to the manufacturer General Electrics. Similar sequences are available from all major MRI manufacturers (SWI by Siemens, SWIp by Phillips, BSI by Hitachi and FSBB by Canon) which produce almost identical images. SWAN produces two images. The signal magnitude is called SWAN and is easily read by humans. The signal phase is called phase image and useful for distinguishing thrombus from artefacts.

Basic Idea SWAN is a susceptibility weighted sequence and is an extension to the T2* weighted imaging. According to General Electrics it is a multi-echo T2* sequence combined with a special reconstruction algorithm [44]. Technical details of the sequence have been presented at the ESMRMB 2008 conference [45] but the publication seems to be confidential and is not available.

Interpretation The intended purpose of the SWAN was to show the veins. Veins appear as fine black lines on SWAN whereas normal brain tissue is white. The black appearance of the veins is due to the magnetic defect caused by the iron atom of the deoxygenated haemoglobin in the veins. Iron in oxygenated haemoglobin is in a different configuration and doesn't produce a magnetic defect. The iron of the clotted blood in the thrombus is in a similar configuration as the deoxygenated haemoglobin and causes a similar defect in the magnetic field. Because of the high local density of iron atoms in the thrombus this defect is larger than the one of the veins and the thrombus appears as a black spot on the image (figure 1.7). Veins are easily



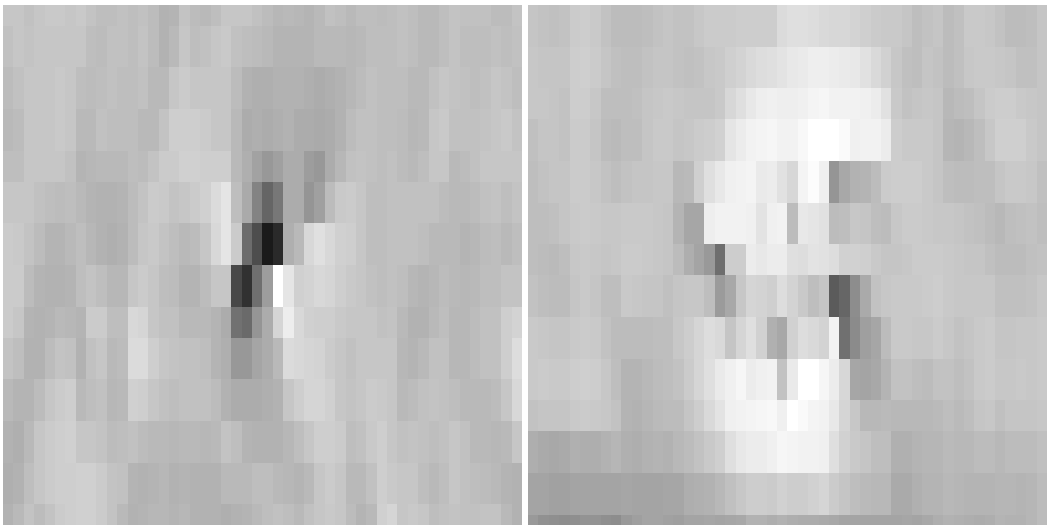
a) b)

Figure 1.7 Slices of a SWAN with a thrombus a) and a calcification b) inside the circle. Just by their appearance they are indistinguishable.

distinguished from the thrombus by their elongated form. Calcifications also appear black on SWAN and can be easily confused with the thrombus but they can be identified by looking at the phase image. Furthermore calcifications tend to appear always in regions of the brain where a thrombus cannot appear, like the mid-sagittal plane.

On the phase image of the SWAN in sagittal view the thrombus appears as a black vertical double cone with the cones touching at the points (figure 1.8 a)). At the touching point it is sourrounded by a white ring. A calcification however appears as the same structure with inversed colors (figure 1.8 b)).

Microbleeds are very small haemorrhages and usually don't provoke any symptoms. They are constituted of a small amount of clotted blood in the brain tissues and are another type of object which can be easily confused with the thrombus. They appear as black dots, identical to the thrombus, but they are usually smaller and not close to the arteries. As these small bleeds are clotted blood they are technically identical to a thrombus. An



a) b)

Figure 1.8 Sagittal view of a thrombus a) and a calcification b) on the SWAN phase image. The pattern of the thrombus is black on top and bottom with a white ring in the middle. The calcification shows the inverse pattern. Due to the image resolution these patterns are smeared but the primary colour and the small amount of the inverse colour around the middle are still visible.

increased number of microbleeds might indicate a vascular disease. Which in turn can contraindicate thrombolytic therapy. Also other haemorrhages appear as large black areas. They are usually larger than the thrombus and easy to spot.

Special care is needed, if a thrombus intersects with a vein. Sometimes the border between thrombus and vein is not clear.

All objects made from clotted blood, *i.e.* thrombus, microbleeds and haemorrhages, are subject to blooming artefacts. They appear bigger than they are in reality. For the thrombus this effect can be partially estimated by looking at the diameter of the artery it is attached to.

1.2.6 Diagnosis Pipeline

All of the above mentioned MRI modalities are used together in clinical stroke diagnosis. Each MRI modality provides an extra piece of information and all together they provide ample information for a treatment decision. All together the MRIs take around 10 minutes to acquire. The first modalities

can be already viewed while the last are taken to speed up the diagnosis. The diagnosis pipeline is as follows:

- NIHSS
  - Provides first estimation about the affected brain region before the MRI
- DWI and ADC
  - Provides size and position of the cytotoxic oedema
  - Affected hemisphere
- FLAIR
  - Provides the size of the core and in comparison with the cytotoxic oedema from DWI an estimate of the penumbra.
  - Allows to distinguish the lesion from the current stroke and old lesions from previous incidents.
  - Rough size of the penumbra can be estimated from collateral arteries
- SWAN
  - Allows to distinguish between ischaemic and haemorrhagic stroke
  - Gives away the thrombus location and size. The lesion location from DWI speeds up the search as the thrombus is nearby.
  - The number of microbleeds helps with assessing the risk for haemorrhages through thrombolysis.
- ToF
  - Verifies the thrombus position through missing parts in the arterial tree.

With this information the treatment decision detailed in section 1.1.2 can be done immediately.

1.3 Automatic Feature Extraction from MRI

Extracting all the information described in section 1.2.6 by hand from the numerous MRI slices is tedious. Furthermore the feature quality strongly depends on the experience of the radiologist interpreting the images. Large parts of the available information are not even considered in the standard scenario because it takes too long to extract them. For example the lesion volume on DWI has been shown to be a predictor for the patients outcome in retrospective studies [1], but manually measuring the lesion volume would take too much time in an acute case.

This is the point where (semi) automatic methods for the analysis of MRI data come into play. A computer could measure the volume of a lesion in a second where a human would need around 30 minutes. There were a lot of tries to develop automatic methods for parts of the diagnosis pipeline above. So far none of them was good enough to be generally accepted in clinical routine.

Beginning with general considerations on working with MRI volumes, this section will give an overview of the existing strategies for automatic feature extraction which have been applied to the stroke MRI modalities.

1.3.1 General Considerations

Working with MRI data-sets comes with some restrictions compared to 1D signals or conventional 2D images. They are true 3-dimensional volumes, with grey values ranging between zero and several thousand for each voxel. On one hand this opens up new ways of analysis, on the other hand it poses some problems.

Data-Set Size One data-set may consist of multiple scans of one patient. These may be from different modalities or different points in time. A stroke data-set may consist of a DWI (which includes the ADC map and the $b = 0$ scan), a FLAIR, a ToF and a SWAN (with the corresponding phase image).

In DICOM format it would take up around 400 to 500 Mb of disk memory. If converted to double precision values for analysis, it gets to the range of 1 Gb per data-set. Considering that standard algorithms [5] may easily take up to 10 times the memory of the input variable, a workstation with at least 16

[5] Standard algorithms in this sense are common image analysis algorithms, which are found pre-implemented in software packages like Matlab.

Gb of RAM is mandatory. For comparing multiple exams at the same time, this is still not enough. Having 32 Gb of RAM or more is recommended for working comfortably with data-sets of this size. With the ongoing technical development this will be probably a non issue in the future, but currently it has to be considered.

There are some strategies to reduce the amount of data. Around 30 to 50% of an MRI scan consists from the air around the head, which can be discarded as it does not carry any useful information. As in stroke the interest lies on the brain and the vasculature, skull stripping algorithms [46] are the method of choice to reduce the volume. Furthermore many diseases of the brain are limited to only one hemisphere, which in turn allows to exlude one half of the data. All together this might cut the memory requirement for an exam by up to 80%.

If a disease is restricted to certain anatomic regions, anatomic labelling [47] can be used to reduce the amount of data further.

Data Quality Most MRI data stems from clinical scanners and hospitals usually have a tight budget and time plan. The MRI machines need to be available to many patients so scan times need to be short. So the scan times are chosen as short as possible which reduces the image quality to the minimum where they are still human readable.

Unfortunately humans are excellent at pattern recognition which means that human readable is actually very difficult to read for a computer. The human brain can correct for an astonishing amount of noise an distortions. Clinical MRI tends to be of low resolution and plagued with artefacts. Thus developing automatic methods on this type of data is a real challenge.

Imbalanced Data A single SWAN has around 70 million voxels. The thrombus captured on this modality has between 200 and 3000 voxels which is less than 0.004% of the volume. Also on MRI of other diseases the actual disease is very small compared to the volume of the scan. This massive imbalance renders most machine learning algorithms useless for application on the raw data. Pre-processing needs to define a region of interest to equalise the balance in order to use machine learning methods.

Noise MRI data is affected by several sources of noise. One part comes from the machine itself due to imperfect magnetic fields and thermal noise in the coils used for generating and measuring the RF pulses. Another part comes from the effect of doing a measurement in the quantum mechanical sense which adds uncertainty to the result. Next up the line is the fact that one voxel is averaged over all the different tissues inside which changes the

result depending on the sample position. All this noise is amplified with the massive amplification needed to measure the tiny NMR signals. Finally the digitalisation and processing of the signals adds another layer of noise.

Gaussian and Poission noise occur in every measurement, while some other types like Rician noise [48] may be mixed in. A sensible correction of the noise can only be done on the signals themselves and is therefore best left to the manufacturers of the MRI machine. Deconvolution techniques on the final images are hopeless, as the point spread function is unknown and location dependent. Also wavelet denoising is more likely to add gradients to the images rather than to remove noise. The best approach might still be not to do denoising at all, or to use a classical median filter. This is viable because the signal of the diseases on MRI are far above the noise level most of the time.

Evaluation of Automatic Methods Even though some frameworks for generating artificial MRI data exist [49], the gold standard for the verification of algorithms are still manually labelled ground truths. Manual labelling might take several hours per patient and needs to be performed by an experienced radiologist. For once this is expensive and results in small training and testing data-sets and it is subject to the problem of intra- and inter-observer agreement [50]. Thus the performance of an algorithm is often rated by comparing the difference between the algorithm and the manual labels to the inter-observer agreement.

1.3.2 Collateral Arteries Segmentation

Currently there is not a single publication concerned with the automatic segmentation of the collateral arteries. Not on FLAIR or any other imaging method.

Manual segmentation of the collateral arteries is not performed either. In the studies which use the collateral arteries the amount of collateral arteries is attributed a grade by experts [51], *i.e.* an indirect rough measurement. On the FLAIR of the stroke data-set available for this work (section 4.1) a sensible detection of the collateral arteries is not possible. Due to the gaps between the slices they are only partially visible. A higher resolution FLAIR would be necessary.

1.3.3 Thrombus Segmentation

No automatic segmentation method for the thrombus on MRI has been published yet. There are a small number of publications about the semiautomatic segmentation of the thrombus on CT and CTA [52, 53, 54, 55, 56]. Only the last three of them are about thrombus segmentation in the brain. They have in common that they need a manual seed to get the algorithm started, *i.e.* they are region growing algorithms and not able to find the thrombus position by themselves.

In Egger et al. [52] the border of the thrombus is segmented manually on one slice. This manual segmentation is then propagated to the next slices with a graph cut algorithm. As the border of a thrombus on SWAN is usually a strong contrast this could be applied to stroke MRI as well. But in cases when veins cross the thrombus it will fail.

In Olabarriaga et al. [53] as well as in the previous method thrombi in aneurysms are segmented. Such a thrombus encloses an artery. Here the artery is first determined by manual start and end points and an additional manual vessel border segmentation on one slice. Then a deformable model is applied to segment the thrombus. In a deformable model a surface is deformed by outer and inner forces. The outer forces are based on the grey-level values of the image and draw the surface to points where a border between a thrombus and other tissue is likely to be. The inner forces punish derivations from a spherical shape, and keep the volume enclosed by the surface compact. Given a proper seed-point, this could actually be used on SWAN, as the inner forces would not allow the segmentation to grow too far into crossing veins.

Qazi et al. [54] investigate the segmentation via thresholds. They conclude that the thrombus can be segmented by user defined thresholds if other areas are excluded. The vital point is that the area of the thrombus needs still to be found by hand. This is not adaptable to MRI as a thrombus on SWAN is not segmentable by a simple threshold.

Once again CTA is used by Santos et al. [55]. Manual seed-points were placed before and after the thrombus and then the region of the artery was obtained by intensity based region growing. The same was done with the contralateral artery and their centerlines were mapped to each other by a 3D B-spline. Then the thrombus was assumed to be the part which was missing in the occluded artery with respect to the contralateral artery. This approach suffers from two major problems. The first is the asymmetry of the

brain vasculature. The diameter on one side does not need to be identical to the other side. Second this only works for partial occlusions, when there is still blood flowing past the thrombus, or the artery is filled from behind via the collateral circulation by the contrast agent. Even on CTA this appears to be only 51% of the cases. It would be difficult to adapt this method to a ToF as the collateral circulation is not fast enough to be detected by ToF.

Finally Riedel et al. [56] used an intensity based region growing which required the user to specify a seed point inside the thrombus. This time CT images were used.

To summarise this there is currently no method to automatically locate the thrombus. If the thrombus is located manually, intensity based region or surface growing can delineate the thrombus on CT and should be able to do the same on SWAN. With the restriction that veins, which are also black on SWAN, might be adjacent to a thrombus and could perturb a region growing approach. The solutions to the thrombus location and segmentation problem which have been developed in this work are discussed in section 4.3.

1.3.4 ToF Arterial Tree Segmentation

Arterial tree segmentation is probably the best explored area of all stroke MRI sequences with the first methods appearing in the early 90's. The available methods comprise a large range of fields from Hessian based filters to machine learning models. A pretty extensive review of methods until 2009 is given by Lesange et al. [57]. Despite the numerous approaches it remains an active area of research where new methods are published regularly [58, 59, 60, 61, 62, 63].

The thrombus is found at the end of an artery where it sits and blocks the blood flow. Intuitively this would give a segmentation of the arterial tree a high values, as only all ends of arteries would need to be checked to find the thrombus. Unfortunately it turns out that on the CHSF stroke data-set (section 4.1) many arteries disappear far before the thrombus. Supposedly the blood flow before the thrombus is sometimes too slow to appear on ToF. So the arterial tree segmented from ToF of clinical quality has no predictive value for the thrombus location.

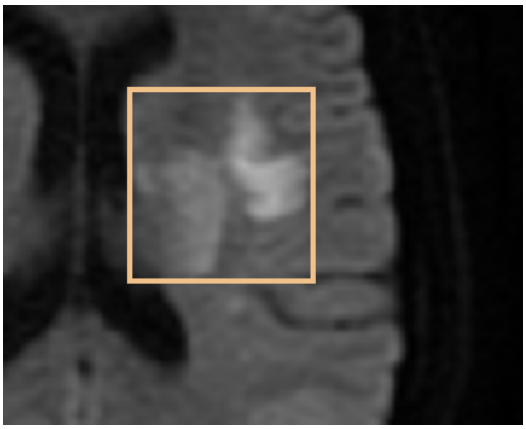


Figure 1.9 A developing lesion on DWI showing the different stages of development. The lower left of the lesion (inside the box) has an intensity like the normal tissue. Towards the right side of the lesion the intensity gradually increases. The right part of the lesion is already well developed and shows strong borders. Most publications only deal with well developed lesions and thus can exploit the high intensity and strong borders. For the hyper-acute phase weak or no borders and low intensities like on the left side make a major part of the lesions and complicate the segmentation task.

1.3.5 DWI Lesion Segmentation

The lesion size on DWI is a popular predictor for patient outcome, and thus numerous attempts have been made to segment the lesion on DWI. For a review of automatic methods available until 2015 please refer to [64].

Stroke lesions are visible on several MRI modalities, depending on the lesion age. The lesion age is usually divided into three categories: acute, sub-acute and chronic. The hyper-acute phase is neglected in most publications because it is very similar to the acute phase, but intentionally added here. Table 1.1 gives the appearance of the lesion on DWI with respect to the normal brain tissue during the different phases.

Between the phases the hyper or hypo-intensities develop gradually. Also the lesion is not one compact block. It consists of sub-regions with varying degree of damage. Thus a lesion which appears acute on one side might already be in an sub-acute state on its other side. On top of that, the degree of hyper-intensity itself in an (hyper-)acute lesion does have a high variance (compare figure 1.9).

For this work only hyper-acute lesions are of interest. Hyper-acute means

Table 1.1 The different phases of stroke. The appearance of the lesion on DWI ADC and FLAIR during the phases is given as well as the period of the phases.

Phase	DWI	ADC	FLAIR	Period
Hyper-acute	hyper-intense	hypo-intense	not visible	hour 0-12
Acute	hyper-intense	hypo-intense	not visible	day 1-7
Sub-acute	hyper-intense	not visible	hyper-intense	week 1-3
Chronic	hypo-intense	hyper-intense	hypo-intense	after week 3

within the first hours after symptom onset, *i.e.* the time when patients arrive at the hospital usually. This is the area where the least publications are available. In the sub-acute phase the crisp hyper-intense signal on FLAIR makes lesion segmentation a task similar to tumour segmentation, for which many algorithms already exist. Likewise in the acute phase the lesion is already well developed with a clear border visible on DWI. The hyper-acute phase on the other hand is very difficult because the lesion is starting to develop and thus the border of the lesion might be anything from a soft gradient to a sharp line. Also the amount of hyper-intensity is far from its peak value. Some of these developing lesions show only a local hyper-intensity on DWI with almost no hypo-intensity on ADC. The only way a neurologist can identify these lesions is by symmetry, *i.e.* there is no similar structure in the other hemisphere.

No published method specially targets the hyper-acute stroke lesion segmentation. However methods for the acute phase work to some degree. In the following two recent methods are presented which demonstrated excellent results for the acute phase and at the same time demonstrate different approaches towards the segmentation problem.

In today's segmentation algorithms convolutional neural networks (CNN) have become prevalent. They are also tried on the lesion segmentation problem. For example almost all submissions for the ISLES 2017 stroke lesion segmentation challenge were based on CNNs with varying architecture, but with rather mediocre results (the best Dice coefficient being reported as 0.31). One example of a well performing architecture is given by [65].

The core idea of their method is to use a CNN (a EDD-net is used) to assign a lesion probability to every voxel. This probability map is thresholded, and the resulting connected components are fed to another neural network (the MUSCLE-net) to classify the components into lesion and artefacts.

The EDD-net is fed with image patches, not with slices. This overcomes

the problem that a lesion makes up only a small part of the MRI volume which causes an under-representation of the lesion in the data-set. In the patches corresponding to an area with lesion, the lesion is well represented. The convolution layers of a CNN loose small details and thus also small lesions. Their very simple but effective strategy to keep those small lesions is to concatenate the original image patches with the extracted feature vectors before the deconvolution layers of the EDD-net. The output is a lesion probability map which is thresholded to obtain a binary mask. The lesion and a couple of artefacts which are similar to a lesion are segmented in this first stage (see section 1.2.2 for a description of the artefacts on DWI).

In the second stage, the binary mask is first decomposed into connected components. For each component a input vector is constructed as follows: multiple image patches of varying size with the component in the centre and one corresponding patch with the probability map from stage one. These input vectors are fed to the MUSCLE-net which determines which of the components are artefacts and lesion.

It is remarkable, that Chen et al. achieved good segmentation results over a large range of lesion sizes and intensity variations. Most other publications avoid to show images of lesions with varying intensity. On the downside, by the nature of the connected component analysis used, their algorithm will be unable to separate lesion and artefacts if they are connected on the image. And they missed some of the very faint parts of the lesion. Nevertheless it is a remarkable work and they achieved unusual high Dice coefficients [66], around 0.61, for small lesions. This CNN architecture is quite complicated but similar results can be achieved with simpler CNNs (see section 4.4).

The second algorithm uses ζ images [67]. A value called ζ is calculated for each voxel:

$$\zeta = \frac{1}{k}\sum_{i=1}^{k}(x - n_i) - \frac{1}{k(k-1)}\sum_{i,j=1}^{k}(n_j - n_i) \tag{1.2}$$

The first term is the average intensity distance between the intensity x of a point and the intensities n_i, $i \in \{1, \ldots, k\}$ of its k nearest neighbours . The second term is the mean intensity distance between the nearest neighbours. ζ is a measure of anomaly of a voxel with respect to its nearest neighbours. A cube of $3 \times 3 \times 3$ voxels is used as the neighbourhood. ζ is used in conjunction with an atlas of normal ζ values. The atlas is constructed as follows:

A number of DWIs of normal patients, without any lesion, are coregistered to a template brain. Then the ζ values for each patient and voxel are calculated. A normal ζ map is created by calculating the median of each voxel across the patients.

To find the lesion of a patient, his DWI is also coregistered to the template brain and his ζ values are calculated. The normal map is subtracted from his ζ image. Regions with a value bigger than zero indicate regions of strong anomalies. After some automatic adaptive thresholding, a binary mask of the lesion is obtained.

This method has a very unique approach to lesion segmentation. It is the only atlas based stroke lesion segmentation, one of the few unsupervised methods and it can easily cope with the artefacts on DWI which tend to be at roughly the same location for most patients. Also lesions which intersect with the artefacts should not be a problem for this method. On the downside, artefacts which are shifted to unusual positions during the coregistration step will be detected as lesions. It is probably unsuited for weak lesions, as their degree of anomaly is also weak compared to the artefacts. On the CHSF stroke data-set (section 4.1) this method is not applicable, as the data-set is too small to create a reliable atlas and no suitable coregistration template is available.

Another approach is based on Fuzzy-C-Means ([68] and [69]). As the Fuzzy-C-Means alone is not capable to delineate the lesions, both authors use the strong edge of the lesion as an additional information. As detailed above, the hyper-acute lesion does not necessarily have a strong edge. So for the hyper acute case this approach is not very interesting. Nevertheless these methods were tried (section 4.4.1) on the CHSF stroke data-set (section 4.1) and delivered bad results, as expected.

Though not being a true segmentation method the approach by Nielsen et al. [70] should be mentioned here. He used a CNN architecture reminiscent of a SegNet [71] to predict the final lesion which would remain after treatment. It is a bit out of the scope of this thesis to predict the development of the lesion, but Nielsen's work assures that CNNs can be used to process the lesion. Also it might be interesting for future works, as the automatic prediction of treatment success for the different treatment methods is planned as a follow up to this thesis.

DWI lesion segmentation is not very different from any other lesion or tumour segmentation task and thus hundreds of methods exist which might be applicable. If the segmentation task is restricted to the hyper-acute phase

the main properties of the lesion used by most algorithms, a strong edge and a strong intensity change with respect to normal tissue, vanish. Thus the number of methods who are promising drops to a few and there is none which was designed for this specific case.

1.3.6 RAPID

RApid processing of PerfusIon and Diffusion (RAPID,[72]) is the only automated feature extraction pipeline which was developed for clinical use. It uses perfusion and diffusion MRI.

From the perfusion MRI the time T_{max} from contrast bolus application to maximum intensity is calculated for each voxel. From the DWI the ADC map is calculated. Both the T_{max} and the ADC map are thresholded with a patient independent threshold. After removal of small components the two binary masks are compared. The mismatch between the two is the perfusion-diffusion mismatch, an indicator for the penumbra.

RAPID provides a method to calculate the volume of the infarct core and the perfusion-diffusion mismatch without the need for a trained radio/neurologist. On the downside it operates with fixed thresholds. This means it cannot adapt to individual variances of the patients. Thus the calculated values are only a crude approximation. But they seem good enough to be worthwhile for the treatment decision.

1.3.7 Summary: Automatic Feature Extraction

For none of the MRI sequences described in section 1.2 a fully automatic feature extraction for the hyper-acute phase of stroke exists in literature. For the thrombus on SWAN and the collateral arteries on FLAIR the literature is a blank page. For the lesion there are a couple of methods for the acute phase which are close enough to try them in an hyper-acute scenario, or to borrow ideas from them. Furthermore ideas can be borrowed from segmentation algorithms for brain tumour MRI or other diseases. The current trend for segmentation algorithms is to use deep learning with CNN architectures. This work is also oriented along these lines and sections 4.3 and 4.4 show how to apply deep learning to the stroke segmentation problem.

Chapter 2

Machine and Deep Learning

The previously described problematic of stroke MRI segmentation has been evoked decades ago but no solutions have been found so far. The traditional image processing approach to image segmentation is obviously not flexible enough for this type of problem. Or it must rather be said the problem is too complex as that humans can engineer the needed filters and selection rules. In the last years image segmentation contests are more and more dominated by a class of supervised data driven methods, deep learning. Deep learning is a subsection of machine learning. Machine learning itself is as old as computers, originating in the need to fit models to measurement data. In machine learning the human designs an algorithm, like a filter for an image, and the numerical values for the algorithm are found, *i.e.* learned, by a computer. This allows to test many strategies for a given problem without having to worry about finding the best numerical values manually. Deep learning refers to using large models where many processing steps are chained together. These "deep" models have proven to be very powerful for a number of tasks. The drawback to machine learning is that the learning part is very resource demanding. This is the reason that in the past only small models have been used and their application to images was mostly impossible due to the limited computational resources available to researchers. With the advent of GPUs for scientific computing the available computational power drastically increased and machine learning with large models was suddenly possible. They proved their power by achieving super-human performance in the ImageNet challenge in 2015 [73]. Ever since deep learning models are the gold standard for automatic image detection tasks.

This chapter aims to introduce machine learning, deep learning and some of the models commonly used for bio-medical data-sets. It starts with some necessary definitions and will then introduce the basics of machine learning.

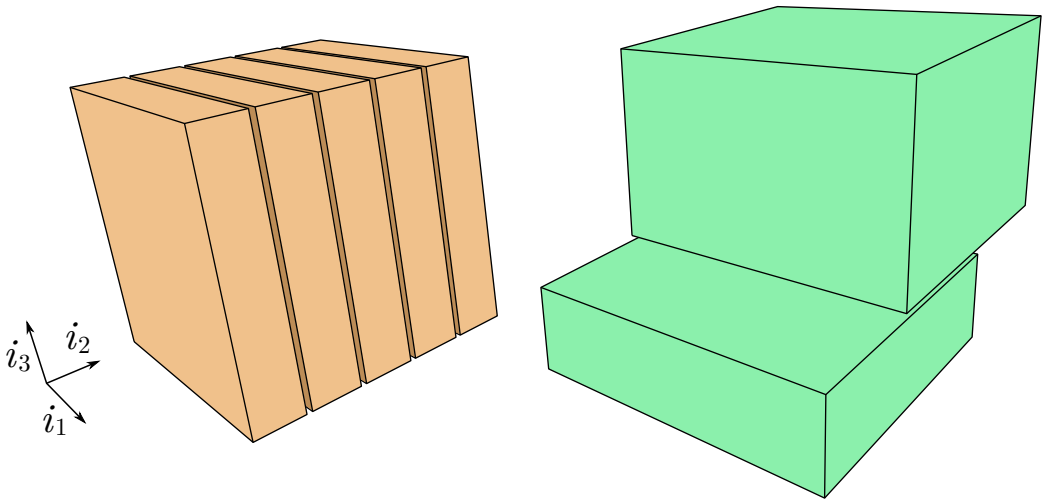


Figure 2.1 Slicing of a volume in five equally sized sub-volumes (left side) and extraction of a larger slice from a volume (right side).

Following this the concept of deep learning will be introduced and building upon that convolutional neural networks and recurrent networks. Finally the Long Short Term Memory (LSTM) [89] based models as well as U-Net variants are presented.

2.1 Definitions

2.1.1 Discrete Intervals

A discrete interval $[a \dots b]$ with $a, b \in \mathbb{N}$ is a sequence of *consecutive* integers which are defined by the intersection of the interval $[a, b]$ with $\mathbb{Z}$:

$$[a \dots b] = [a, b] \cap \mathbb{Z} \tag{2.1}$$

2.1.2 Working with Volumes

Volumes in the sense of this work are hypercubes of an N-dimensional space with a cartesian grid. Each position on the grid is attributed a value. Thus the volume is a finite set of values (voxels) $v(i_1, i_2, \dots, i_N) : \mathbb{N}^N \to \mathbb{R}$ which are indexed through their position on the grid, i.e. by N indices $i_1, i_2, \dots, i_N$ with $i_k \in [1 \dots n_k]$. The n_k are called the axis sizes of the volume. The

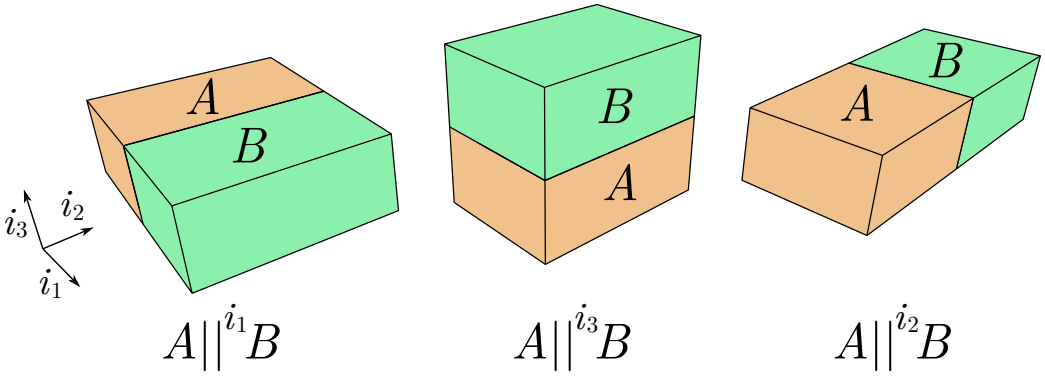


Figure 2.2 Concatenation of two volumes A and B along different axis, forming a new larger volume.

volume V can be written in a tensorial notation as:

$$V_{i_1 i_2 \ldots i_N} = \{v(i_1, i_2, \ldots, i_N) | i_k \in [1 \ldots n_k], k \in [1 \ldots N]\} \tag{2.2}$$

There are two major operations for manipulating volumes. The first is called slicing which means to extract a sub-volume S from the volume by restricting the range of one or more indices (figure 2.1). This can be written as:

$$S_{j_1 j_2 \ldots j_N} = (V_{i_1 i_2 \ldots i_N})_{i_k \in [a \ldots b]} \qquad a, b \in [1 \ldots n_k] \tag{2.3}$$

Note that the sub-volume S is still in the same N-dimensional space as V. But the axis size n_k corresponding to the index j_k of S is now smaller, *i.e.* $j_k \in [1 \ldots b - a + 1]$ and $n_k = b - a + 1$. It may even happen that $n_k = 1$.

The second operation is concatenation which means that two volumes V and T are joined together (figure 2.2). This requires that for the axis sizes n_k of V and the axis sizes m_k of T holds:

$$n_k = m_k \; \forall \; k \neq l \tag{2.4}$$

where l is the concatenation axis. The concatenation itself is defined as:

$$V||^l T = C_{j_1 j_2 \ldots j_N} = \begin{cases} V_{i_1 i_2 \ldots i_N} \text{ with } i_k = j_k & \text{for } j_l \leq n_l \\ T_{i_1 i_2 \ldots i_N} \text{ with } i_k = j_k, \; i_l = j_l - n_l & \text{for } j_l > n_l \end{cases} \tag{2.5}$$

This definition of volume is aimed at working with real world volumes such as MRI images or photographies. For these volumes one of the axis has a special

name: the feature or channel axis and the remaining axes are called are the image axes. For example a digital colour image usually has three axes. Two axes define the image (image axes), and the third axis gives the red, blue and green channel of the image (channel axis). Processing this image results in extracting features for each pixel in the image. This creates a new image with the image axes as in the original image and the channel axis becomes a feature axis. The terms channel and feature are interchangeable. Which one is used depends on the speaker having a signal processing (channels) or machine learning (features) point of view. By convention in this work the feature channel is always indexed by i_N, i.e. the last axis.

2.1.3 Convolution

In general the convolution of two functions $f, g \in \mathcal{L}^1(\mathbb{R}^n)$ is defined as:

$$(f * g)(x) = \int_{\mathbb{R}^n} f(x-t)g(t)dt \tag{2.6}$$

This can be restricted to the discrete convolution of two functions $f : A \to \mathbb{R}, g : B \to \mathbb{R}$ with $A, B \subset \mathbb{N}^N$:

$$(f * g)(x) = \sum_{y \in B} f(x-y)g(y) \tag{2.7}$$

Note that A will be a finite set for all applications in this work. On finite sets it can happen that $(x-y) \notin A$. In this case there are two approaches: the first one is to restrict the sum to values of y such that $(x-y) \in A$. This is the "valid" discrete convolution:

$$(f * g)(x) = \sum_{\{y \in B | (x-y) \in A\}} f(x-y)g(y) \tag{2.8}$$

The second option is to set $f(x-y) = 0 \ \forall (x-y) \notin A$. This is commonly called zero padding and allows $(f * g)$ to be defined on the same set A as f. Therefore it is often referred to as the "same" (size) convolution:

$$(f * g)(x) = \sum_{y \in B} f(x-y)g(y) \ \ | f(x-y) = 0 \ \forall (x-y) \notin A \tag{2.9}$$

Throughout this work zero padding is used, if not stated otherwise.

The common use case for convolutions in machine learning is their application to volumes, as defined above. Some naming conventions have been established for this. f becomes the input volume (or image) with axis sizes n_k and g is the convolution kernel with axis sizes m_k. The x and y become indices for the volumes f and g. The channel or feature axis has a special role. By convention g is the identity along the feature axis and the output of the convolution can have multiple features in contrast to the definition above. The number of input features is n_N, given by f. The number of output features is n_o and a parameter of the convolution operation. This is achieved by packing multiple convolutions into one operation, one for each output feature. A from above is now the set of valid indices for the input volume f and is defined as $A = \{(i_1, i_2, \ldots, i_N) | i_k \in [1 \ldots n_k], k \in [1 \ldots N]\}$. As there are multiple output features the kernel g gets an additional axis and thus the indices of g are in the set $B = \{(j_1, j_2, \ldots, j_{N+1}) | j_k \in [1 \ldots m_k], k \in [1 \ldots N+1], m_N = n_N, m_{N+1} = n_o\}$. The resulting volume of the convolution operation has the same axis sizes as f except for the feature axis. The convolution with zero padding written in the terms of volumes becomes:

$$a_k = i_k - j_k + \lceil m_k/2 \rceil \quad k \in [1 \ldots N-1] \tag{2.10}$$

$$a_N = j_N \tag{2.11}$$

$$f_{a_1 a_2 \ldots a_N} = 0 \quad \forall (a_1, a_2, \ldots, a_N) \notin A \tag{2.12}$$

$$(f * g)_{i_1 i_2 \ldots i_N} = \sum_{j_1 j_2 \cdots j_N} f_{a_1 a_2 \ldots a_N} \; g_{j_1 j_2 \ldots j_N i_N} \tag{2.13}$$

$\lceil \rceil$ denotes the ceiling function. In many publications the kernel size is split into a image and feature part, *i.e.* the convolution operation defined by equation 2.13 would be described as a convolution with a $m_1 \times m_2 \times \cdots \times m_{N-1}$ kernel and m_N input features/channels and m_{N+1} output features/channels. The number of input features is determined by the size of the input image f, *i.e.* known, and thus almost always omitted. This way of speaking is also adapted in this work.

2.1.4 Maximum Pooling

An important operation for contemporary neural networks is maximum pooling [74]. It is defined for a volume $f : A \to \mathbb{R}, A = \{(i_1, i_2, \ldots, i_N) | i_k \in$

$[1 \dots n_k], k \in [1 \dots N]\}$ and a set $B = \{(i_1, i_2, \dots, i_{N-1}, 0) | i_b \in [-K_b \dots K^b], b \in [1 \dots N-1]\}$ called window where either $K_b = K^b$ or $K_b = K^b - 1$ with $K^b \in \mathbb{N}$:

$$maxpool(f, B)(x) = \max_{y \in B} f(x - y) \tag{2.14}$$

Note that this operation looks for the maximum in a neighbourhood given by B along the image axis. Unlike the convolution the channels are not mixed in this operation. Often the maximum pooling operation is used for down-sampling the volume by restricting x. This restriction is called **striding** with stride $s \in \mathbb{N}$ and A is restricted to $A' = \{(i_1, i_2, \dots, i_N) | i_k \in [1, 1 + s, 1 + 2s, \dots 1 + n_s s], n_s = \lceil n_k / s \rceil - 1, k \in [1 \dots N]\}$ with $\lceil \rceil$ being the ceiling function. The strided maximum pooling operation is then:

$$x' = 1 + sx \tag{2.15}$$

$$maxpool(f, B, s)(x) = \max_{y \in B} f(x' - y) \tag{2.16}$$

The strided maximum pooling reduces the size of the input image by only considering every sth entry along all image axes and discarding all others. The concept of strides can be used for convolution operations as well and with fractional strides can even be used for up-sampling [75].

2.2 Machine Learning

Machine learning is, in the widest sense, building machines which are able to learn a task by observation of examples. Currently these machines are computer programs and the algorithms are closely linked to statistics and aimed at modelling probability densities which describe some data. Any of the current models obey the same principle.

A model $f(x, \theta)$ has a set of parameters θ which determine the model's behaviour. The model has the task to map an input x to an output y such that $y = f(x, \theta)$. Consider x as a random variable drawn from an unknown probability distribution $P(X)$. To construct a perfect model the conditional probability distribution $P(Y|X)$ would be needed. But for real applications it is unknown and only a set of inputs $\mathcal{X}$ and their corresponding outputs $\mathcal{Y}$

are known. The goal of machine learning is to estimate θ so that the predictions $\hat{y} = f(x, \theta)$ are as close as possible to the true values y. In an ideal case not only the known examples $\mathcal{X}$ are modelled well, but the model really models (has learned) the unknown probability distribution. In another point of view machine learning solves an optimisation problem on a given data set $(\mathcal{X}, \mathcal{Y})$, though not all machine learning problems can be mapped to an optimisation problem.

There are numerous ways of constructing the model $f(x, \theta)$ and a large part of machine learning research is dedicated to improving the models or finding better ones. The second big topic in machine learning is how to estimate (=learn) θ.

The most used approach for parameter learning is to calculate $\hat{y} = f(x, \theta)$ for a small set of specific examples and an initial random θ and then compare $\hat{y}$ to y with the help of a loss function $L(y, \hat{y}) = L(y, f(x, \theta))$. The loss function measures the similarity of $\hat{y}$ and y and by convention smaller values of $L(y, \hat{y})$ indicate higher similarity. Then by evaluating the first and sometimes the second derivatives of $L(y, \hat{y})$ with respect to θ a small correction Δ is calculated such that $L(y, f(x, \theta + \Delta)) < L(y, f(x, \theta))$. This is called gradient descent and although many alternatives have been tried, up to date it is the most efficient method for training neural networks and many other machine learning methods.

2.2.1 Loss Functions

When speaking about a loss almost never the value of the loss function for a single sample is meant. Rather the loss function is calculated either for all available samples or a subset (mini-batch) and the mean of all the individual values is called "loss". This denomination is adapted here, although the mean is omitted in the definition of the loss functions. There are many loss functions in use for machine learning but only the most important two for segmentation tasks are covered here.

Cross Entropy The cross entropy can be directly derived from the maximum likelihood estimation. Be $q(x, \theta)$ the probability densitiy described by the model and θ the model's parameters. Furthermore suppose a set of samples $\mathcal{X}$ from the true probability distribution of $p(x)$.

Then the parameters θ_{ML} which give the best approximation of $\mathcal{X}$ by the model is given by the maximum likelihood estimation:

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}} \prod_{x \in \mathcal{X}} q(x, \theta) \tag{2.17}$$

Taking the logarithm and scaling doesn't influence θ_{ML} and turns the product into a sum, which is easier to handle:

$$\begin{aligned}
\theta_{ML} &= \underset{\theta}{\operatorname{argmax}} \sum_{x \in \mathcal{X}} \log(q(x, \theta)) && (2.18)\\
&= \underset{\theta}{\operatorname{argmax}} \sum_{x \in \mathcal{X}} \frac{1}{|\mathcal{X}|} \log(q(x, \theta)) && (2.19)\\
&= \underset{\theta}{\operatorname{argmin}} \sum_{x \in \mathcal{X}} \frac{-1}{|\mathcal{X}|} \log(q(x, \theta)) && (2.20)\\
&= \underset{\theta}{\operatorname{argmin}} \sum_{x \in \mathcal{X}} \frac{1}{|\mathcal{X}|} \log\left(\frac{1}{q(x, \theta)}\right) && (2.21)\\
&= \underset{\theta}{\operatorname{argmin}} E\left[\log\left(\frac{1}{q(x, \theta)}\right)\right] && (2.22)
\end{aligned}$$

Where the expected value in the last line is already the cross entropy $H(p, q)$. The dependency on p is shown below. Thus minimising the cross entropy is equivalent to a maximum likelihood estimation of the model's parameters. In practise this works very well and therefore cross entropy is possibly the most used loss function. More commonly the cross entropy is written as the sum of entropy $H(p)$ and Kullback-Leibler divergence $D(p||q)$:

$$\begin{aligned}
H(p, q) &= E\left[\log\left(\frac{1}{q(x, \theta)}\right)\right] && (2.23)\\
&= E\left[\log\left(\frac{1}{p(x)}\right)\right] + E\left[\log\left(\frac{p(x)}{q(x, \theta)}\right)\right] && (2.24)\\
&= H(p) + D(p||q) && (2.25)
\end{aligned}$$

Dice Loss The Dice coefficient [66] has long been a measurement of the similarity of two sets A and B, defined as:

$$Dice = \frac{2|A \cap B|}{|A| + |B|} \tag{2.26}$$

It has the nice property to be bounded in $[0, 1]$ which is good for numerical stability. A value of 1 means the two sets are identical and 0 means no overlap. This can be easily turned into a loss function, the Dice loss:

$$L_{Dice} = 1 - \frac{2|A \cap B| + \alpha}{|A| + |B| + \alpha} \tag{2.27}$$

Now 0 means identity of A and B to make it a loss functions which can be minimised. And α is a small constant which serves to avoid division by 0 if the sets are empty by chance. The Dice loss directly measures segmentation quality and therefore automatically takes care of class imbalance. In problems with severe class imbalance, such as segmenting very small objects in medical images the Dice loss may outperform the cross entropy. An examination of the convergence properties and a generalised version of the Dice loss is found in [76].

2.2.2 Gradient Descent

The most common class of methods for minimising a loss function is gradient descent. Classical gradient descent looks at the loss calculated over all available samples. Then the derivative $\Delta = \frac{\partial}{\partial \theta} L(y, \hat{y})$ is calculated and the parameters are updated as $\theta_{new} = \theta - \eta\Delta$ with the learning rate η. This is repeated until convergence. Unfortunately this method does not always find a good minimum and may be stuck at saddle points or spurious minima. Many variants of gradient descent have been developed to circumvent this weakness or to speed up convergence. A contemporary overview of gradient descent variants used in deep learning is given by [77].

Stochastic Gradient Descent Calculation of the loss over all available samples tends to provide a smooth error surface which can cause problems with saddle points. If there are many similar samples it wastes performance on recalculating almost the same gradient over and over. The stochastic gradient descent calculates the loss only for a single example between parameter updates.

This leads to very different gradients for each update. While this may slow down convergence it also helps with leaving saddle points or local minima.

Mini-Batch Gradient Descent In this version the gradient is calculated over a small subset of all samples, the "mini-batch". The gradient is smoother between updates than with stochastic gradient descent but still varies enough to avoid the problems of classic gradient descend. Using mini-batches for gradient descend is well established and used in conjunction with any other improvements to gradient descend.

Momentum A ball rolling down a slope has a momentum which stabilises its descent and makes the ball jump over small crevices without major changes to its direction. On that idea a momentum term has been added to the gradient descent [78], where a part of the last update $-\eta\Delta_{last}$ is added to the current update:

$$\theta_{new} = \theta - \eta\Delta - \tau\eta\Delta_{last} \tag{2.28}$$

With a momentum parameter τ. Ripples in the error surface have less influence and regions with gradients close to zero can be traversed quickly.

Adaptive Moment Estimation (Adam) Adam [79] has been used for training all neural networks in this work. It keeps an exponentially decaying average of the first and second moment of the gradient in memory. This helps quickly traversing planes with flat gradient like momentum and unlike momentum gets slowed down on steep slopes. Thus it is less likely to overshoot a minimum following a steep descent and has less problems with oscillations around minima.

2.3 Deep Learning

Building a house from bricks is much simpler than building a house in one production step. This is the core idea to deep learning. Instead of having one highly complex model $f(x, \theta)$ with millions of parameters θ, which is very difficult to train, simpler steps are chained together. Dividing the model into a sequence of smaller models, $f(x, \theta) = f_1(f_2(f_3(\ldots, \theta_3), \theta_2), \theta_1)$, has the advantage that at each stage f_i of the model there are only few parameters θ_i to consider. This leads to easier parameter estimation. At the same time the chaining of (non-linear) models produces a highly complex model with immense modelling capacities. So the whole model is complex, but the individual steps are kept rather simple and thus relatively easy to learn.

The building blocks $f_i(x, \theta_i)$ can be basically any machine learning method. But in the last years the usage of neural networks as building blocks has become so prominent that many people confuse deep learning with deep neural models or deep convolutional neural networks. The reason of the high interest in neural models is very simple: they are easy to train, fast at evaluation and many software packages are available which allow building these models with ease. It doesn't mean that these are the best possible models, but for now they deliver the best known results.

2.4 Neural Networks

Neurons in a human brain inspired this machine learning method. Still it is very far from being a model of the human brain. A neuron receives electrical impulses (action potentials) from incoming connections of other neurons. The connections to the neuron are via chemical synapses which transforms the incoming action potentials into postsynaptic potentials. When the sum of all postsynaptic potentials exceeds a threshold voltage (ca. -55 mV) the neuron itself fires, i.e. sends an outgoing action potential. As well known a network of these neurons (a brain) can solve an amazing number of tasks. Reason enough to copy this concept.

2.4.1 Perceptron

The perceptron aims at modelling a single neuron. The incoming connections form an input vector $\vec{x}$ where each entry in $\vec{x}$ is one connection. Each incoming connection in a neuron is connected via a synapse which can modulate the "strength" of the connection. This is reflected in the perceptron by multiplying $\vec{x}$ by a weight vector $\vec{w}$. The threshold voltage becomes a threshold value b. So the output y of a perceptron is

$$y = \Theta(\vec{w}^T \vec{x} + b) \tag{2.29}$$

where Θ is the heaviside step function. The original perceptron [80] is a linear classifier and therefore rarely used nowadays.

Often more than one output value is desired. For this the weight vector is replaced by a weight matrix W:

$$\vec{y} = \Theta(W\vec{x} + \vec{b}) \tag{2.30}$$

For modelling non linear functions the heaviside step function is replaced by a nonlinear function called activation function F:

$$\vec{y} = F(W\vec{x} + \vec{b}) \tag{2.31}$$

This version of a perceptron is called a fully connected or dense layer. Every entry of the output $\vec{y}$ is influenced (connected) by every entry in $\vec{x}$. Popular choices for the activation function are sigmoids (e.g. the logistic function or the hyperbolic tangent) and recently the rectified linear unit (RELU) [81].

For good performance in nonlinear tasks the perceptron is extended to the multilayer perceptron. Here n perceptrons are chained together:

$$\vec{y} = F_n(\vec{b}_n + W_n F_{n-1}(\vec{b}_{n-1} + W_{n-1} \dots F_1(\vec{b} + W_1\vec{x}))) \tag{2.32}$$

For $n = 2$ (and this holds for higher n as well) the universal approximation theorem [82, 83] states that the multilayer perceptron can approximate continuous functions on $\mathbb{R}^k$ up to arbitrary precision. For higher precision the number of parameters (or rows) in W_1 needs to be increased.

There are two ways of speaking about multilayer perceptrons. In one each intermediate result of an operation F_i is called a layer. In the other each operation F_i is called a layer, which will also be the convention used in this work. The word layer comes from the graphical representation of the perceptrons and is used for the operations in any neural network. Note that in general there are many possible operations apart from a dense layer.

2.4.2 Neural Networks

Apart from the multilayer perceptron there are many ways to build a network. They can be divided into two major types:

Feed Forward Networks The multilayer perceptron is the prime example of a feed forward network. In a feed forward network the input $\vec{x}$ is processed in steps such that each step only uses results of past steps back to $\vec{x}$. Thus the network can be represented as an acyclic graph which starts with $\vec{x}$ and ends with $\vec{y}$.

Recurrent Networks In contrast to the feed forward network a recurrent neural network (RNN) can be a cyclic graph. A step can be dependent on the result of a future step, or even its own result. Also network models with a memory (i.e. $\vec{y}$ depends not only on $\vec{x}$ but also on all other inputs the model has seen in the past or future) fall under the definition of a recurrent network. This comes from the fact that "memory" can be represented by a self loop in the model.

Both types are constructed as a sequence of operations, also called layers, with simple to calculate gradients. This structure makes training with gradient descent (section 2.2.2) quite easy. Via the chain rule for derivatives the derivative of the loss can be calculated analytically for each layer. This results in the gradient of each layer being dependent on the derivatives of the following layers. This can be seen as the error being propagated "backwards" through the network in order to calculate the gradients of the individual layers. Hence the name "backpropagation" for training of neural networks with gradient descent.

There are many choices for layers. The most popular ones are dense layers (equation 2.31), convolution layers (equation 2.13) and maximum pooling layers (equation 2.14).

2.4.3 Convolutional Networks

Networks built exclusively from fully connected layers can in theory model any continuous function. This also works for images if they are linearised in order to be represented as a vector $\vec{x}$. In practise two problems arise which make the use of fully connected layers undesirable.

The first is that fully connected layers are not **translation invariant**. This is easily understood with an example of image classification. Assume that handwritten numbers have to be detected on an image. Once the network

has learned to detect the number "2" in the left upper corner of the image it will do so quite well. But if the number "2" is moved to any other part of the image it will no longer be recognised as the number "2". Thus the training set needs to contain an example of the number "2" in any possible position on the image. For handwritten numbers this is doable, but tedious. For a set of natural images it may not be possible to acquire images with all the possible translations of an object and with high resolution images (i.e. many possible translations) the size of the training database would become unmanageable.

The second problem is that the number of parameters grows with the input and output size. For image segmentation tasks the input is an image and the output is a (segmented) image as well. An image of resolution 512×512 has 262144 pixels. With $\vec{x}$ and $\vec{y}$ of that size the matrix W of a single perceptron would already have 68 719 476 736 entries. When working with 32 bit numbers this would correspond to 256 Gb of memory requirement. So the models are too large for contemporary hardware.

The solution to both problems is the usage of convolutions. A convolutional layer is written as

$$y = F(W * x + b) \tag{2.33}$$

where the convolution is defined as the discrete convolution in equation 2.13. The number of parameters in W now becomes a free choice as the matrix multiplication is gone and convolution is defined for two signals of arbitrary length. This solves the hardware issues, as the models size can be arbitrarily scaled. Also the input and output no longer needs to be a vector but can be a matrix or any higher order tensor.

The translation invariance comes from the fact that the convolution kernel W is the same for every possible position of the input. So once the network learns to recognise a number on one position on the image it automatically will recognise it at any position.

The introduction of convolutions to neural networks suddenly made them viable for image processing and classification tasks. Since then convolutional neural networks (CNN) have become the most powerful and widely used machine learning tool and have made tasks possible which were deemed to difficult before.

But the use of convolutions comes with a cost. In a fully connected layer the

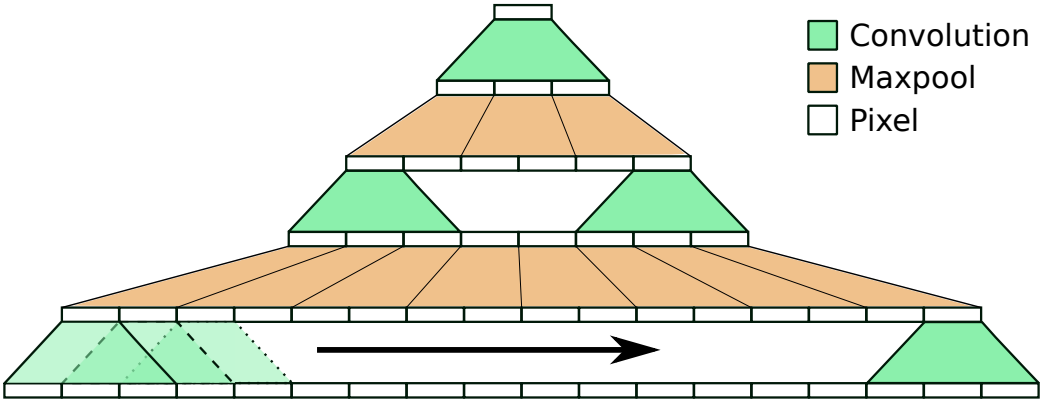


Figure 2.3 An input image (bottom) is reduced in resolution with alternating convolution and maximum pooling layers. The convolution layers have a kernel size of 3 and the maximum pooling layers a window size of 2 and a stride 2. With 5 operations the final pixel on the top has a receptive field of 18. Note that the receptive field grows exponentially with the number of maximum pooling layers. This example also nicely demonstrates the pyramidal structure emerging from multi-resolution techniques

output has access to all inputs. In a convolution an output pixel has only acces to the part of the input given by the size of the convolution kernel. This part of the input which is available to the final layer in a network is called **receptive field** . The goal of all CNN architectures is to increase this receptive field until it is large enough for the task at hand. For most image analysis tasks this is the entire input image. Using large convolution kernels is not possible as the computational complexity and amount of parameter grows quadratic with the kernel size. Therefore the kernel sizes are set to 3×3 or 5×5 in most architectures.

But the receptive field can be linearly increased by stacking convolutional layers. With two 3×3 layers following each other, the last one can "see" a 4×4 neighbourhood in the input layer. By stacking 3×3 layers, the receptive field increases by one per convolution. Which means a lot of convolutions must be stacked to have a receptive field as large as a reasonable input image. The increase in receptive field per convolution can be drastically higher when using maximum pooling layers [74] with a stride of 2. The stride induces a subsampling of the image to a lower resolution. Figure 2.3 illustrates the effect of alternating convolution and pooling layers on the receptive field. With such a pyramidal structure of sufficient depth the last convolution will have access to a (down-sampled) version of the whole image. For image segmentation tasks this down-sampled image is once again up-sampled to the

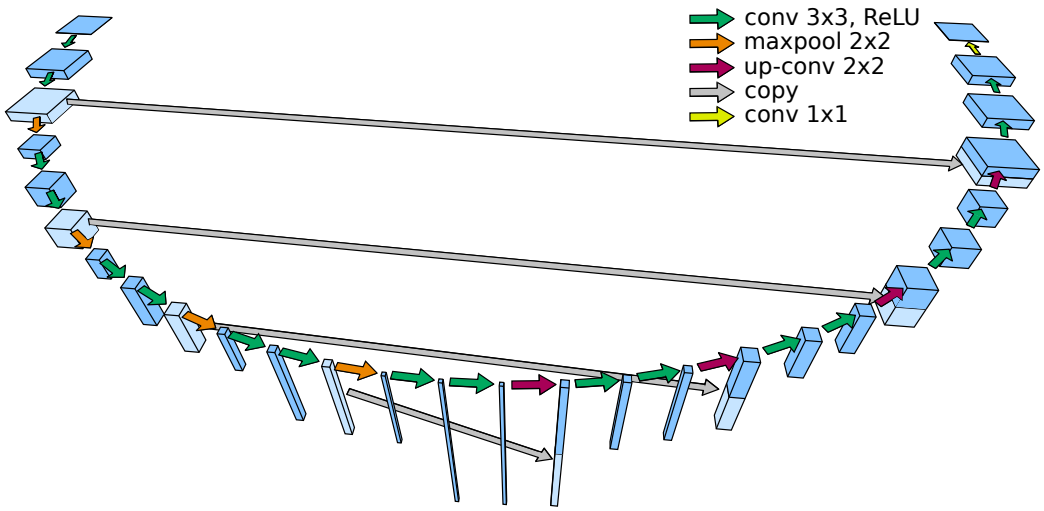


Figure 2.4 The U-Net architecture. Arrows represent operations and cubes represent feature maps where the height of the cube stands for the number of feature maps and the width and depth of the cubes for the size of the feature maps. It is clearly visible that the U-Net uses a pyramidal structure for feature extraction, i.e. a sequence of feature maps of decreasing resolution. Then from the smallest resolution there is an inverse pyramidal structure of up-sampling steps which lead to a pixel-wise segmentation in the end.

original size in an inverse pyramid pattern. This concept has been known for a long time [84] and is widely applied in todays CNNs. Tsung-Yi Lin et al. [85] have a good overview of CNN architectures which use such pyramid/ multiresolution techniques.

2.5 Models

From the multitude of neural network models only some are useful for segmentation tasks. The existing architectures which inspired this work or were actually used are explained in this section.

2.5.1 U-Net

Three years after the breakthrough of convolutional neural networks for image classification tasks [86] the U-Net [87] set a new standard for the segmentation of bio-medical images. It was not the first to be applied to bio-medical

image segmentation tasks, but it combined multiple ideas which made it outperform its precedessors. It is de facto the archetype of a convolutional neural network for image segmentation.

As any classic convolutional neural network the U-Net subsequently downsamples the input image in a compressing branch. At this stage features at multiple resolutions are extracted (figure 2.4 left side). The resolution is halfed with a maximum pooling layer and the number of features at each resolution is doubled with a convolutional layer.

This is followed by an upsampling branch which increases the resolution back to the input space (figure 2.4 left side). Notably transposed 2D convolutions are used for upsampling, i.e. the upsampling operation is learned. The other notable part is the use of skip connections, (figure 2.4 grey arrows, the copy operation). The idea behind the skip connections is to use the features which were extracted in the downsampling branch again during the upsampling in order to preserve fine-grained details. This is one of the key-points which allows this model to produce segmentations with pixel precision.

Finally the classification layer uses 1×1 convolutions adopting the principle of the all convolutional network [88]. This avoids fully connected layers and their drawbacks all-together.

The U-Net has been successful in a number of bio-medical image segmentation tasks and is well accepted. As it combined all characteristics of a modern convolutional neural network it is a good reference model.

2.5.2 Long Short Term Memory (LSTM)

Early recurrent networks had a problem remembering information for long periods, like several thousand time steps. Hochreiter et al. [89] introduced a special memory cell which was able to retain information for extended periods of time. The LSTM can read and write to its memory. More important this memory is never passed through an activation function. This effectively combats the vanishing gradient problem and makes training this model very stable.

The original LSTM works with a series of input signals $\vec{x}_t$. It has a so called hidden state $\vec{h}_t$ and cell state $\vec{c}_t$ of the same size as $\vec{x}_t$. The cell state $\vec{c}_t$ is the models memory. The hidden state $\vec{h}_t$ is the model's prediction of $\vec{x}_t$.

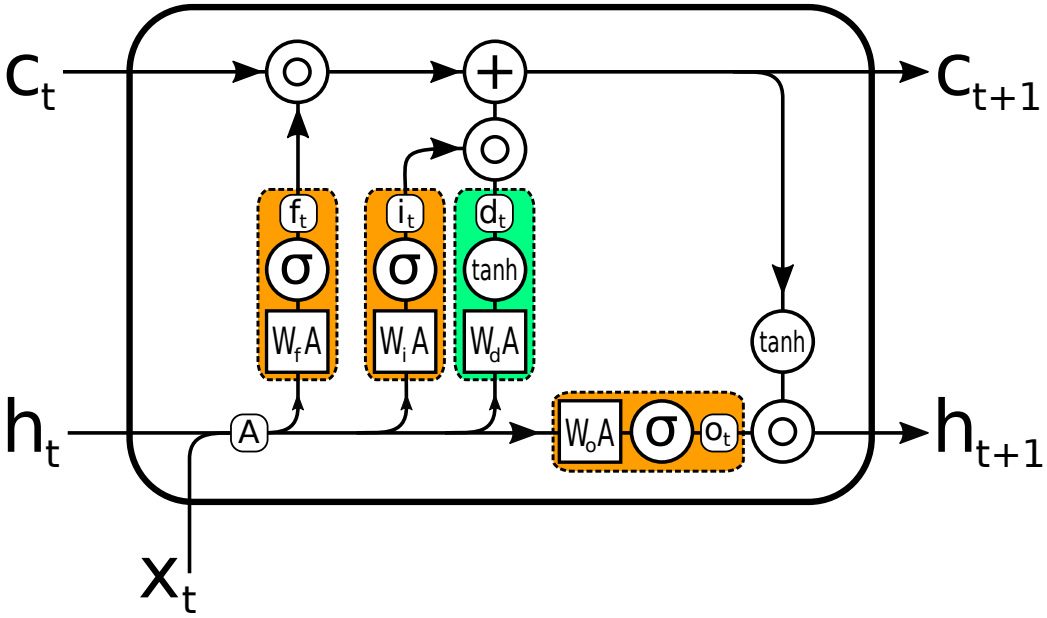


Figure 2.5 Visualisation of the operations defining the LSTM (equations 2.34 to 2.40. The gates which decide which part of the information to pass on are orange. Green is the update to the memory cell.

With the Hadamard product $\circ$ the LSTM is defined by the equations

$$\vec{A} = \vec{h}_t || ^1 \vec{x}_t \tag{2.34}$$

$$\vec{f}_t = \sigma(W_f \vec{A} + \vec{b}_f) \tag{2.35}$$

$$\vec{i}_t = \sigma(W_i \vec{A} + \vec{b}_i) \tag{2.36}$$

$$\vec{o}_t = \sigma(W_o \vec{A} + \vec{b}_o) \tag{2.37}$$

$$\vec{d}_t = \tanh\left(W_d \vec{A} + \vec{b}_d\right) \tag{2.38}$$

$$\vec{c}_{t+1} = \vec{f}_t \circ \vec{c}_t + \vec{i}_t \circ \vec{d}_t \tag{2.39}$$

$$\vec{h}_{t+1} = \vec{o}_t \circ \tanh(\vec{c}_{t+1}) \tag{2.40}$$

Where σ is the logistic function and the W are weight matrices and $\vec{b}$ biases. What looks like a complicated model at first is very straightforward with the proper explanation. The basic idea is that the model takes the input $\vec{x}_t$ and the previous prediction of the current input $\vec{h}_t$, updates its internal memory $\vec{c}_t$ to $\vec{c}_{t+1}$ and then makes a new prediction $\vec{h}_{t+1}$ based on $\vec{c}_{t+1}$, $\vec{h}_t$ and $\vec{x}_t$.

For convenient notation the $\vec{x}_t$ and $\vec{h}_t$ are concatenated into a single vector $\vec{A}$ (equation 2.34). The forget gate $\vec{f}_t$ (equation 2.35) decides which part of the memory should be either erased or kept. Thanks to the logistic function $\vec{f}_t$ contains values between 0 and 1. Thus in the left term of the memory update (equation 2.39) every entry of the memory $\vec{c}_t$ is multiplied by 1 (i.e. kept), by 0 (erased) or by some value $]0, 1[$ (i.e. the memory is gradually lost). On the same idea the input gate $\vec{i}_t$ (equation 2.36) decides which part of the memory $\vec{c}_t$ needs to be updated. The memory update $\vec{d}_t$ (equation 2.38) is calculated from $\vec{A}$ and passed through a hyperbolic tangent. This activation function serves to squash the input between -1 and 1 which is necessary to avoid numerical instabilities by growing numbers in $\vec{c}_t$. With $\vec{f}_t$ and $\vec{d}_t$ the memory is updated to $\vec{c}_{t+1}$. Finally the prediction $\vec{h}_{t+1}$ (equation 2.40) is made by reading a part of the memory. Which part of the memory is read is determined by the output gate $\vec{o}_t$ (equation 2.37). The hyperbolic tangent in equation 2.40 once again restricts the output range to the interval $[-1, 1]$. The data flow in the LSTM is illustrated in figure 2.5.

The original LSTM could have multiple parallel memory cells $\vec{c}_t$, but as in practice mostly only one memory cell is used the description of the LSTM was limited to one $\vec{c}_t$. All the gate functions (equations 2.35 to 2.37) are fully connected layers (equation 2.31) with a sigmoidal activation function thus the LSTM is a recurrent neural network.

Also the role of $\vec{h}_t$ is not strictly fixed to be a prediction of $\vec{x}_t$. In fact it can be any series of predictions which is connected to the input series $\vec{x}_t$. For example if $\vec{x}_t$ was the number of people who entered (or left) a building in the last hour then $\vec{h}_t$ could be the current number of people inside the building (with an appropriate scaling so it fits the output range $[-1, 1]$).

Peephole LSTM An important modification of the LSTM is the addition of "peepholes" [90]. Each gate can also use the memory $\vec{c}_t$ for its decision. Thus what is written to and read from the memory is also dependent on the memory itself. Peepholes are added by modifying equation 2.34 to

$$\vec{A} = \vec{c}_t||^1\vec{h}_t||^1\vec{x}_t \tag{2.41}$$

whereas the remaining formulas are not modified. In many circumstances this improves the performance of the LSTM.

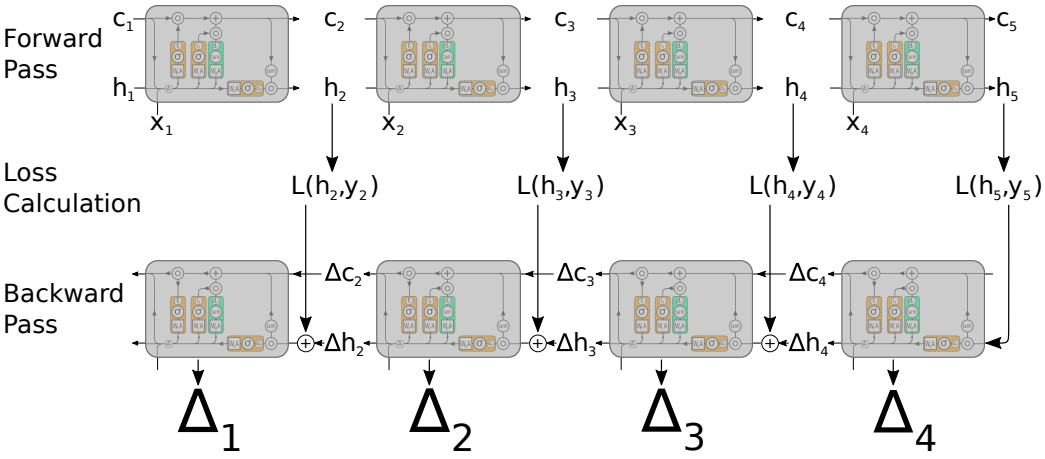


Figure 2.6 Backpropagation through time illustrated at an example LSTM with four inputs. The LSTM is unfolded in time which makes it look like an ordinary feed forward network. But each gray box is indeed the same LSTM with the same parameters. In the forward pass (first line) all hidden and cell states are calculated. Then the losses for all pairs of hidden state and the corresponding ground truth y are calculated. Then the errors are backpropagated as if it were a feed forward network, producing a gradient Δ_t for each step t. The final gradient Δ for the parameter update is $\Delta = \sum_t \Delta_t$.

2.5.3 Backpropagation Through Time

Recurrent neural networks (RNNs) can be trained by **backpropagation through time**. For this the network is unfolded, which means the loops in the network are turned into feed forward connections by replicating the networks structure. An unfolded LSTM is shown in figure 2.6, row one. After unfolding the network is represented as a directed acyclic graph where the original RNN is repeated over and over. In this unfolded representation most RNNs are very deep networks, which makes it clear why RNNs are counted as deep learning methods. But in contrast to a feed forward network many layers share their weights.

The unfolded network is trained with backpropagation (i.e. gradient descent, section 2.2.2), as if every layer had its own set of weights. The final gradient for a parameter is the sum of all individual gradients which where calculated for that parameter throughout the unfolded network. For example if a parameter w appears m times in the unfolded network it will receive m different gradients Δ_t. The gradient for w is then $\sum_t^m \Delta_t$.

The calculation of the gradient by backpropagation through time is illus-

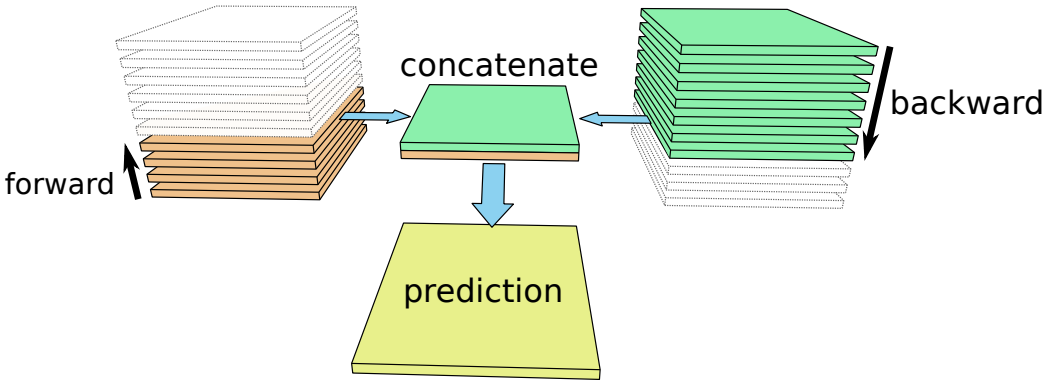


Figure 2.7 The principle of the bidirectional RNN illustrated for a bidirectional CLSTM working on a 3D volume. The forward model (left side) processes the volume slice by slice until the target slice. The backward model (right side) starts from the top and processes the slices in reverse order until the target slice. The output of both models for the target slice is concatenated (middle) and then passed through a classification layer to produce the prediction. The forward network sees only the data from the bottom to the target slice and the backward network sees only the data from the top to the target slice. The concatenated network outputs then contains information about the whole data volume and thus allows predictions which take the whole volume into account.

trated in figure 2.6 at the example of an LSTM with four inputs. Here the main advantage of the LSTM can be seen, the error which is backpropagated through $\vec{c}_t$ never passes through an activation function. So all errors passed into the memory persist back to the first input and are not diminished. This allows the LSTM to be correctly trained on inputs with long range dependcies.

For real time series with thousands of inputs the unfolded network might become so large that it results in hardware problems. In these cases the backpropagation is truncated at a specified depth (**truncated backpropagation through time**). The loss at time t is only propagated back to time $t-k$ where $k < t$. Although this is then only an estimate of the gradient it is still sufficient for training in many cases.

2.5.4 Knowing the Future

While the LSTM is well suited to prediction tasks on time series, sometimes the knowledge about future events is necessary for a correct prediction. For example if traffic jams should be predicted along a road then t is not the time

but the location on this road. Suppose $\vec{x}_t$ would be the traffic density then a good prediction for the risk of a traffic jam might need the information of the traffic density up and down the road from the loaction t. With the LSTM model above only the $\vec{x}_{t-k}$, $k \in \mathbb{N}$ would be known to the memory of the model. So the term future is relative to t and means the following data-points. Of course the following/future data points need to be already known to be integrated into the prediction. According to [91] there are two strategies to incorporate knowledge of future events into an LSTM model.

Bi-directional RNN First is the bidirectional RNN approach [92]. For this approach the entire signal must be known. Two LSTM models are trained in parallel, one on the normal input series (forward) and the other one on the inversed input series (backward), starting with the last input then the second last and so on (compare figure 2.7). Thus for every t there are two hidden states $\vec{h}_{1,t}$ and $\vec{h}_{2,t}$ from the two models available. $\vec{h}_{1,t}$ contains information only about the past and $\vec{h}_{2,t}$ information only about the future. Together they have the information about the whole signal and the final prediction $f(\vec{h}_{1,t}, \vec{h}_{2,t})$ is made using both hidden states. This method has the drawback that two models need to be trained and thus the amount of parameters and training time are doubled.

Delayed Input The second approach is delaying the signal by a delay τ. This means that the hidden state $\vec{h}_t$ gives not a prediction for time t but a prediction for the past time $t - \tau$. Consequently the prediction made for $t - \tau$ using $\vec{h}_t$ has access to all future events which passed in the time window $[t-\tau, t]$. Thus a bit of the future is available for predictions. But the drawback is that for an input time series of length N only $N - \tau$ predictions can be made effectively excluding the end of the time series from predictions. This can be circumvented by extending the input time series with τ fake inputs of zero or constant value. But when τ becomes large the model needs to deliver a prediction reconstructed from memory alone while dealing with a repeated zero input. Imagine that τ becomes larger than the size of the memory $\vec{c}_t$, then predicting τ steps at the end of the signal becomes nearly impossible. Thus the method of delaying the input, while being resource friendly, is restricted to delays which are short compared to the whole input length. In contrast to the Bi-directional approach the delayed input can be applied to real-time problems. For example a camera stream could be processed in "almost" real time with a delay of some seconds.

2.5.5 Convolutional LSTM

Introduced by Shi et al. [93], the convolutional LSTM (CLSTM) brought the power of LSTMs to the world of images. Instead of a series of values (a signal), it takes a series of images as input, where each image may have multiple channels as well. In the interest of a more general definition in this work the CLSTM is defined for volumes using the notation from section 2.1.2.

The CLSTM corresponds to a LSTM with peepholes where the dense layers have been replaced with convolutions. Thus the input no longer needs to be a vector but can be a more general series of volumes $X^t_{i_1 i_2 \dots i_N}$. Of course the hidden state $H^t_{i_1 i_2 \dots i_N}$ and the cell state $C^t_{i_1 i_2 \dots i_N}$ become volumes too. There is little change to the equations, they become

$$A = C^t_{i_1 i_2 \dots i_N} ||^N H^t_{i_1 i_2 \dots i_N} ||^N X^t_{i_1 i_2 \dots i_N} \tag{2.42}$$
$$f_t = \sigma(W_f * A + \vec{b}_f) \tag{2.43}$$
$$i_t = \sigma(W_i * A + \vec{b}_i) \tag{2.44}$$
$$o_t = \sigma(W_o * A + \vec{b}_o) \tag{2.45}$$
$$d_t = \tanh\Big(W_d * A + \vec{b}_d\Big) \tag{2.46}$$
$$C^{t+1}_{i_1 i_2 \dots i_N} = f_t \circ C^t_{i_1 i_2 \dots i_N} + i_t \circ d_t \tag{2.47}$$
$$H^{t+1}_{i_1 i_2 \dots i_N} = o_t \circ \tanh\big(C^{t+1}_{i_1 i_2 \dots i_N}\big) \tag{2.48}$$

Note that the equations 2.43 to 2.46 now have convolution operations. Special care has to be taken with the biases which are still vectors with length n_N. The addition of the vector happens along the channel axis i_N as

$$(C_{i_1 i_2 \dots i_N} + \vec{b})_{i_N = t} = (C_{i_1 i_2 \dots i_N})_{i_N = t} + b_t \tag{2.49}$$

So each element b_t of the vector $\vec{b}$ is added as a scalar to the slice $(C_{i_1 i_2 \dots i_N})_{i_N = t}$. The weights W are now convolution kernels.

Apart from the change to convolutions and that the input can now be an arbitrary tensor the convolutional LSTM works exactly the same as the LSTM with peepholes. But the convolutions introduce a new drawback to this method, the receptive field (section 2.4.3) is restricted to the size of the convolution kernels. This severely cripples the CLSTMs capabilities to capture spatial patterns, as seen in the experiments in section 3.1.3. Therefore a CLSTM is never used separately but always multiple are used in a variety of architectures.

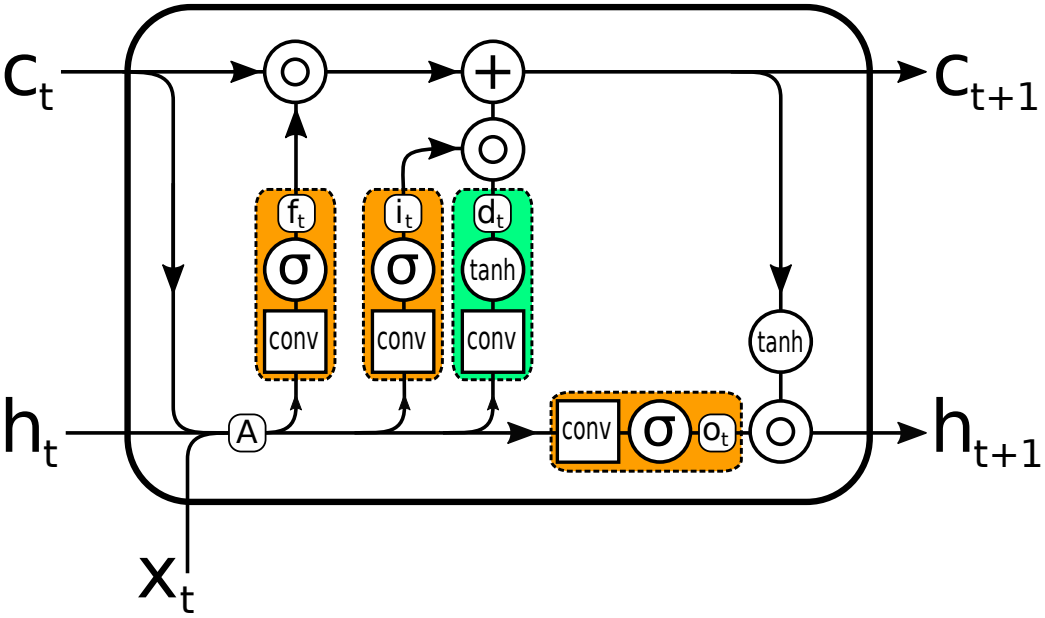


Figure 2.8 Visualisation of the operations defining the CLSTM (equations 2.42 to 2.48. It is almost identical to the LSTM (figure 2.5 apart from the added peephole connection (the left downward arrow) and the change from matrix multiplication to convolutions.

Architectural Principles for CLSTM

There are three types of architectures in which CLSTMs are used. All have in common that multiple CLSTMs have to be trained.

CLSTM Stacks The inventors of the CLSTM [93] used a stack of four CLSTMs. Stack means that the output of one CLSTM becomes the input of a following CLSTM, very much like convolutional layers are stacked to create deep networks. Through this stacking the receptive field is increased as it is in CNNs.

Multi-Resolution Pyramid These architectures are integrating the CLSTM as a building block into multiresolution architectures similar to the U-Net (section 2.5.1). For example the convolution layers in the downsampling path can be replaced by CLSTMs [94, 95] or the copy operations of the U-Net are replaced by entire CLSTMs [96]. These architectures fully profit from the massive increase of the receptive field through a multi resolution method and at the same time can treat input series as compared to single inputs of an U-Net.

Multi-Directional CLSTMs Especially for 3D volumes one axis is treated as the "time" axis of the input, i.e. the input is a series of 2D images. Along the time axis the CLSTM can capture everything thanks to its memory, *i.e.* along the time axis the receptive field is infinite. To remedy the small receptive field of the CLSTM, each of the three axes of the 3D volume is used as "time" axis. This yields three different 2D input series, one for each axis of the 3D space. On each of the three input series CLSTM models are trained, and thus for each of the three directions there is a model which knows the entire input. In a bidirectional approach (section 2.5.4) on each of the input series two CLSTMs are trained. This approach is for example used by [97] and [98], though the latter are using not exactly a CLSTM but a closely related LSTM variant.

Chapter 3

New Approaches

The problem of stroke MRI segmentation has proven too difficult to be solved with established segmentation methods from literature (see section 4.3). Thus it was necessary to improve and adapt these methods. This research has led to new insights about neural network architectures. It resulted in two new building blocks for neural networks, the transfer block and the logic block and a couple of architectures which use them. Notably this solves the inherent problems of the CLSTM. The CLSTM suffers from a small receptive field and current workarounds lead to huge models which are cumbersome to train and thus the CLSTM is rarely used. The logic block removes the receptive field restriction and the leads to the logic LSTM which provides all benefits of the LSTM while working on volumes like the CLSTM. At the same time a logic LSTM uses less parameters than a comparable CLSTM in contrast to previous works [93, 94, 95, 96, 97, 98] which used considerably more parameters than the CLSTM. This chapter describes the technical details of these innovations as well as the experiments to validate them.

3.1 Transfer Block

3.1.1 Motivation

In recent years convolutional neural networks have become the preferred tool for pattern recognition. One big part of their success lies in the translation invariance. The other part is the fact that through a clever choice of architecture the network is able to make decisions considering the whole image.

Section 2.4.3 describes how the receptive field can be increased by alternating

convolutional and maximum pooling layers. This leads to pyramidal network structures (see figure 2.3). CNNs have been very successful for image tasks, but how do they actually work?

The common way to think about what happens in these networks is to think of them as a series of feature extractors. Features are extracted subsequently at multiple resolutions. Also the extracted features have to be robust in the sense that they convey useful information for the following resolutions, *i.e.* they survive the down- and up-sampling operations. The features in the middle of the feature pyramid, after down-sampling and before up-sampling, are very descriptive for the task at hand. Here the close similarity to auto-encoders becomes apparent.

Another way of looking at the inner workings of these models is that information is transferred across the image, so that each pixel in the final layer has access to the whole image. Or at least a big part of it. Thus the receptive field can be seen as the information intake area. Stacking convolutional layers is currently the only way to transfer information across the image which is applied in practise. From a historical viewpoint, fully connected layers can do the same job but come with the main problem that they are not translation invariant. That means they need to learn to detect an object for every possible location of that object separately. For decent image sizes the number of weights to learn becomes so big that contemporary hardware cannot handle this. Thus fully connected layers have been relegated to classification layers at the end of pyramids where the number of pixels has become small, as for example in the Alex Net[86].

But even models with stacks of convolutions regularly report having several ten million variables [87, 86]. So there arises the question: are there alternative ways to transfer information across an image, i.e. to increase the receptive field?

The new transfer block is one alternative which has the advantage that it uses less parameters than a comparable feature extraction pyramid. As it is so small the transfer block can also be a building block within more sophisticated architectures.

3.1.2 Transfer Block Definition

The proposed "transfer block" is a new building block for neural networks architectures. It provides an alternative to pyramidal feature extractors and uses significantly less parameters. Thus the "transfer block" allows to build

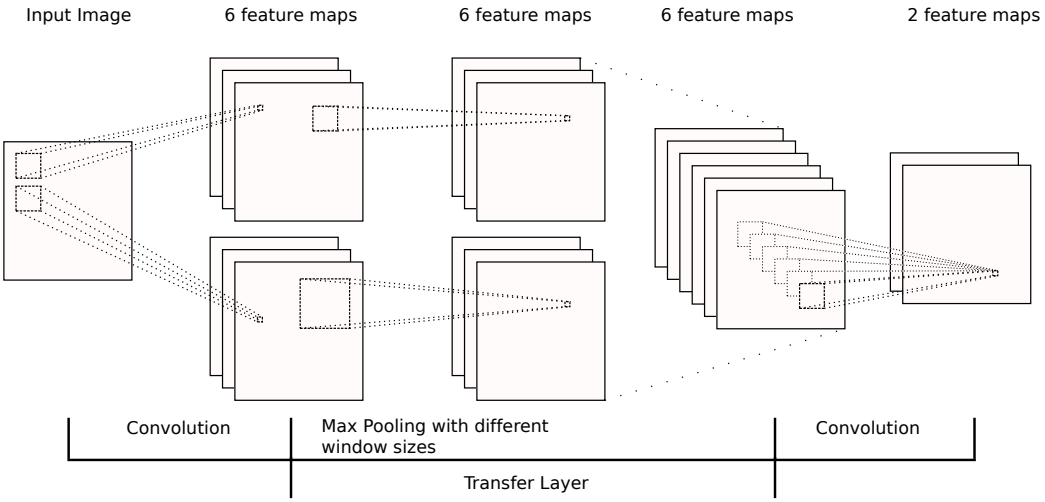


Figure 3.1 An example of the whole transfer block where $m = 3$ and the list of window sizes L contains only two entries. The input image is passed through a convolution layer which produces 6 feature maps. Each of the 6 feature maps is then subject to a maximum pooling operation with stride one, where 3 are processed with the first window size in L and the other 3 are processed with the second window size in L. The 3 each feature maps for the two maximum pooling operations are displayed side by side here for visualisation purposes. In the definition the 6 feature maps are always one stack.

more lightweight models compared to contemporary CNNs.

The Transfer Block consists of three layers. The core part is the transfer layer which is enclosed by two conventional convolutional layers. This transfer layer builds upon the idea that information needs to be transferred across the image in order to have a maximum receptive field for the final neuron. This is achieved through multiple maximum pooling operations [74] with different window sizes.

Be $maxpool((I_{i_1 i_2 i_3})_{i_3=t}, B)$ the maximum pooling operation from equation 2.14 but restricted to the feature map t of the 2D image I with window size B and stride 1. Suppose a list of window sizes L where $L(j)$ is the window size number j with a total number of n_w window sizes. Be the multiplicity m an integer number and suppose that the number of channels of the input image $n_3 = mn_w$. The input to the transfer layer is an image I of size $n_1 \times n_2 \times n_3$ and the output is of the same size. On each of the input channels a single maximum pooling operation is performed and the result is the output for the corresponding channel. Each available window size in L is used on m distinct channels. To formalise this define an index $u = jm + k$ with $k \in \{1, \ldots, m\}$

and $j \in \{1, \ldots, n_w\}$. Then the channel u of the output image O of the transfer layer is :

$$(O_{i_1 i_2 i_3})_{i_3=u} = maxpool((I_{i_1 i_2 i_3})_{i_3=u}, L(j)) \tag{3.1}$$

The transfer layer is preceded by a convolutional layer with a kernel of size $w \times w \times n_{in} \times n_{start}$. The output goes through another convolutional layer with a kernel of size $w \times w \times n_{start} \times n_{out}$. The choice of n_{out} is rather uncritical and almost any value works well. We use $n_{out} = n_{start}/m$. Both convolution layers are followed by ELU [99] as activation function.

Each of the three layers has its purpose in this architecture. The initial convolution layer learns m features for each window size, which means the learned features are specific for the neighbourhood size given by the window size. The transfer layer distributes these features in a neighbourhood corresponding to the respective window sizes. The final layer has access to the features of the neighbouring pixels, which allows access to spatial orientations of features. This is very important as without the final layer the exact borders of objects cannot be determined due to the indifferent nature of the maximum pooling operation towards positions.

The parameters to choose for the transfer block are the window sizes L and the multiplicity m. To transfer information across the whole image, the largest window size should be two times the input image size, i.e. $2 \max\{n_1, n_2\}$. The smallest window size should be 1 in order to preserve the finest details. Choosing the window sizes as powers of 2, i.e. $1, 2, 4, 8, 16$ etc, seems to be a good choice for keeping the model small and gives decent results for most cases. For more complicated tasks it may be necessary to use more window sizes, mostly around the size of the relevant objects. For example if a task contains many objects with a size of around 40 pixels it may be useful to use window sizes of $38, 40, 42$. The multiplicity m decides how many features are learned for each window size and can be used to tune the network's size. If it is chosen too small, the network may be unable to learn the task. But even for complex problems $m \leq 10$ turned out to be reasonable. At last there is also the kernel size w. Using kernel sizes of $w = 3$ seems to be very popular with most CNN architectures. Setting $w = 5$ for the transfer block leads to a faster convergence, larger kernels than that seem to provide no improvement. Initialising the weights of the convolution kernels with the method of Kaiming et al. [100] works reasonably well. An example of the transfer block with specific configuration is given in figure 3.1.

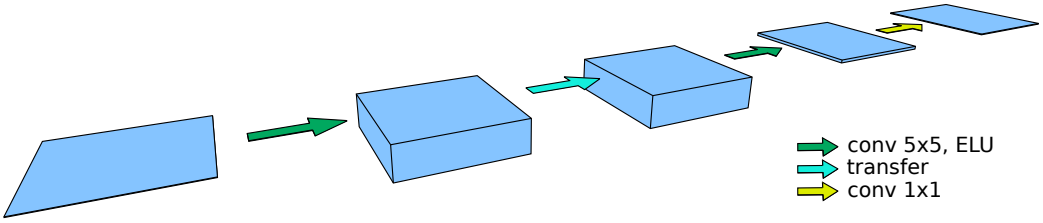


Figure 3.2 The Transfer-Net architecture. In contrast to the U-Net all feature maps have the same resolution, only the number of feature maps changes in each step. The information transfer across the image is done by the transfer layer instead of a multi-resolution pyramid.

With only two convolutional layers the transfer block is a very small architecture in terms of learnable parameters. The multiple resolution steps of feature extraction pyramids are translated to multiple window sizes. As the maximum pooling has no learnable parameters, the transfer block is very lightweight compared to a feature extraction pyramid. The power of the transfer block is demonstrated in the following experiments, where it achieves comparable performance to a feature extraction pyramid while having only a small percentage of a feature pyramid's learnable parameters.

3.1.3 Experimental Validation

In two experiments architectures containing the transfer block are compared to the U-Net model described in section 2.5.1. The U-Net was chosen because it is the archetype of modern convolutional networks used for bio-medical image segmentation tasks and achieved good performance in many applications. To prove that the tasks are not too simple and that the transfer layer is responsible for the good results, a simple reference network with two convolutional layers is included. For evaluating the results not only the loss is used but also the Dice coefficient [66] which is the standard measure for segmentation quality.

Architectures

All of the following network architectures are implemented in Pytorch [1]. Convolution layers are chosen to keep the output size the same as the input size by padding the input image with zeros. Also all architectures are followed

[1] pytorch.org

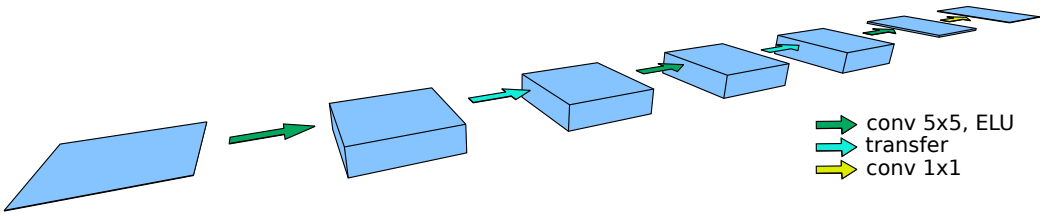


Figure 3.3 The Double Transfer-Net architecture. With this architecture we show how simple transfer layers can be chained together. Chaining transfer layers results in a deeper architecture. Our second experiment shows that the deeper version is indeed better at complex tasks.

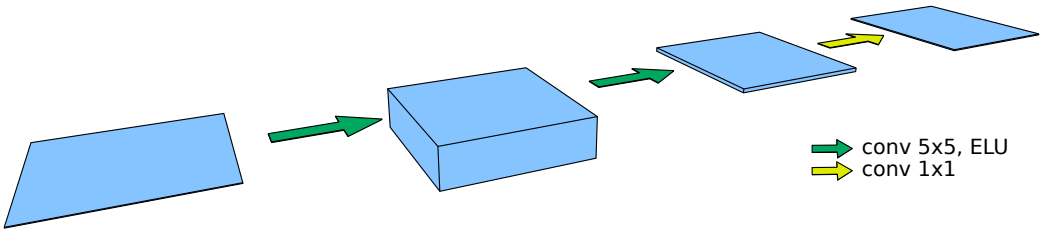
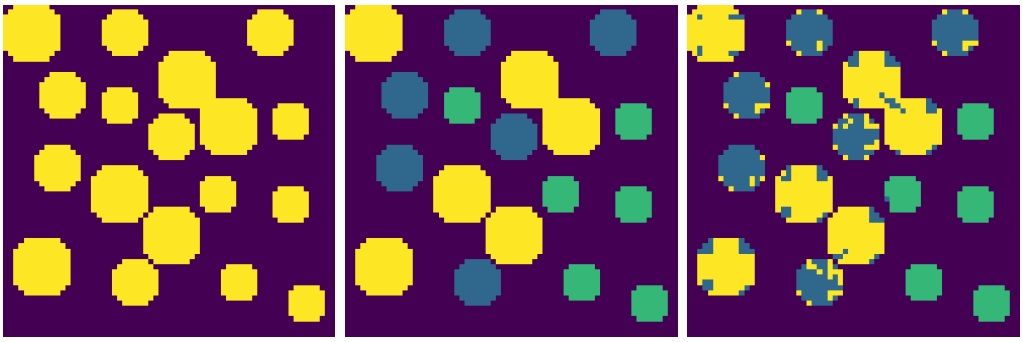


Figure 3.4 The Reference-Net architecture represents the most basic convolution neural network architecture. It suffers greatly from a small receptive field. As expected our first experiment showed that it has trouble with objects that are larger than the convolution kernels. This weakness can be overcome either by adding a single transfer layer to the architecture (Transfer-Net), or by adding so many convolutional layers that the chained convolution kernels extend the receptive field to the whole input image (U-Net).

by a softmax layer (not explicitly stated below) and cross entropy is used as loss function. The loss is minimised using Adam [79]. The best learning rate for each architecture has been determined experimentally.

U-Net The U-Net [87] (figure 2.4) consists mainly of a feature extraction pyramid followed by an expanding path which up-samples the features to the space of the original image. A special feature of the U-Net are its skip connections which allow it to preserve fine grained details. The implementation used here is almost identical to the original paper, with only two differences. First for the input image more than one channel is allowed. Second the crop operation is not implemented, thus the output image of this U-Net implementation has the same size as the input image. This change has no effect on the number of parameters, but increases the execution time slightly as the final images are a few pixels larger. The original U-Net works with five different resolutions and is called U-Net 5 in the following. A smaller variant which works with four different resolutions is also included in the experiments



a) b) c)

Figure 3.5 One example of the circles validation-set. The binary image a) had to be segmented according to the ground truth b). The U-Nets and the Transfer-Nets perfectly segmented the image and delivered exactly b). The result of the R-Net is shown in c). As the receptive field of the R-Net is not large enough to capture the largest circles entirely it has trouble discerning the corners of the large and middle circles as they are identical.

and called U-Net 4.

Transfer Net (T-net) Here the transfer block from above is included in the simplest possible architecture (figure 3.2). It has a first 5×5 convolutional layer, then a transfer layer, a second 5×5 convolutional layer and then a 1×1 convolutional layer for the classification, as recommend by [88] . The window sizes are 1 for the smallest and then $2^k + 1, k \in \{1, 2, 3, \dots\}$ for the larger window sizes until the largest possible for the given image. Odd window sizes are used because Pytorch's maximum pooling function currently has problems keeping the image size constant with even window sizes.

Double Transfer Net (DT-Net) This sample architecture shows how to chain transfer blocks together in order to create deeper architectures (figure 3.3). It starts with a 5×5 convolutional layer followed by a transfer layer, another 5×5 convolutional layer then another transfer layer and a final 5×5 convolution layer followed by a 1×1 convolution for classification. All other parameters are identical to the transfer Net.

Reference Net (R-Net) This is a reference architecture (figure 3.4) which is used to demonstrate that it is the transfer layer which is responsible for the results and not the fact that two convolutional layers are chained together. It is identical to the Transfer Net but without the transfer layer. Thus it starts with a 5×5 convolutional layer followed by a second 5×5 convolutional layer and then the 1×1 convolutional layer for classification. The parameters for the convolutional layers are always chosen identical to the Transfer Net.

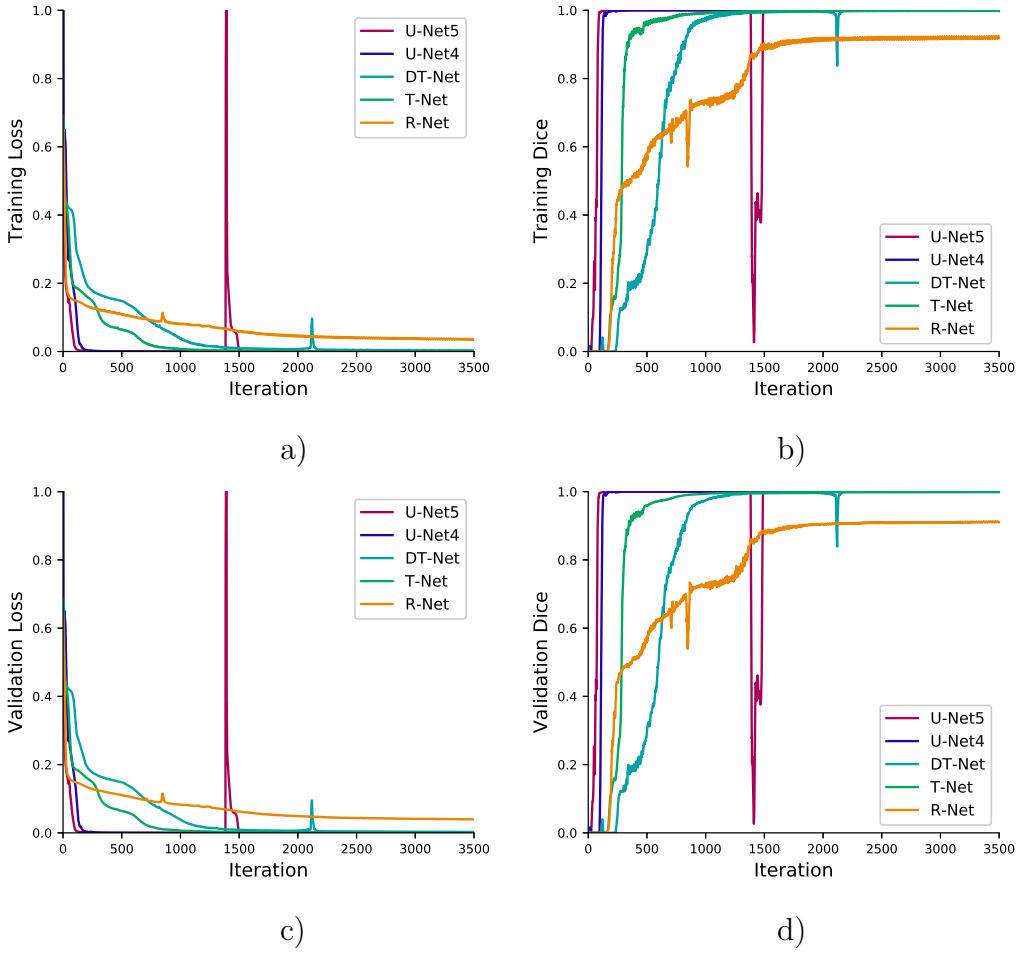


Figure 3.6 Development of the loss a), c) and the Dice coefficient b), d) on the training and validation set for the circles data-set.

Synthetic Data

As a simple example of multi class segmentation binary images of size 64×64 with circles with a radius of 3, 4 and 5 pixels are generated. The circles are placed randomly and do not overlap but may touch each other. Each size of circle is a class for segmentation. The training-set was 20 images and the test-set 10 images. All five architectures were trained on this data-set with the configuration as in table 3.1.

As expected the U-Nets learn this task quickly and the DT-Net and T-Net show no problems as well (see figure 3.6). The Reference Net already fails this task (compare figure 3.5), as its receptive field is smaller than the size of

the circles and thus it cannot measure the surface or diameter of the circles. This first test shows already that the information transfer via the transfer layer works well.

Table 3.1 Network configurations for the circles experiment and their final Dice score which is identical for training and test-set.

Model	Kernel Size	Parameters	Learning Rate	m	Dice
U-Net 5	3	31030788	0.001	-	1.0
U-Net 4	3	7696388	0.002	-	1.0
DT-Net	5	136679	0.0005	10	1.0
T-Net	5	14109	0.002	10	1.0
R-Net	5	14109	0.005	10	0.92

3.1.4 BraTS

For the second experiment a real data set which is renowned for its difficulty is chosen. The brain tumor segmentation (BraTS) [101] challenge is a recurring challenge attached to the MICCAI Conference. Each year the segmentation results become better, but the problem is an ongoing research. For this experiment the high grade glioma part of the BraTS 2017 data-set[2] is used. It contains multi-modal MRI of 210 patients which were manually segmented by experts, i.e. a ground truth is available. On these image three different classes have to be segmented from the background. The enhancing tumor, the necrotic and non-enhancing tumor and as a third class the peritumoral oedema. This makes it an ideal real data-set for supervised learning of a multi-class segmentation task. It was also chosen to provide a publicly available reference.

Training Data Preprocessing The data-set is divided into 100 patients for learning, 4 for validation and left 106 aside as test set. This partition was chosen because the first idea had been to compare the different models by their Dice on the test set. But as the validation set was not used to determine the end of the training there is technically no difference between the validation and test set. As the purpose of this experiment is only to compare the models, the validation and training set are sufficient. So the test set was never used in the end. Four different MRI modalities (compare figure

[2] www.med.upenn.edu/sbia/brats2017.html

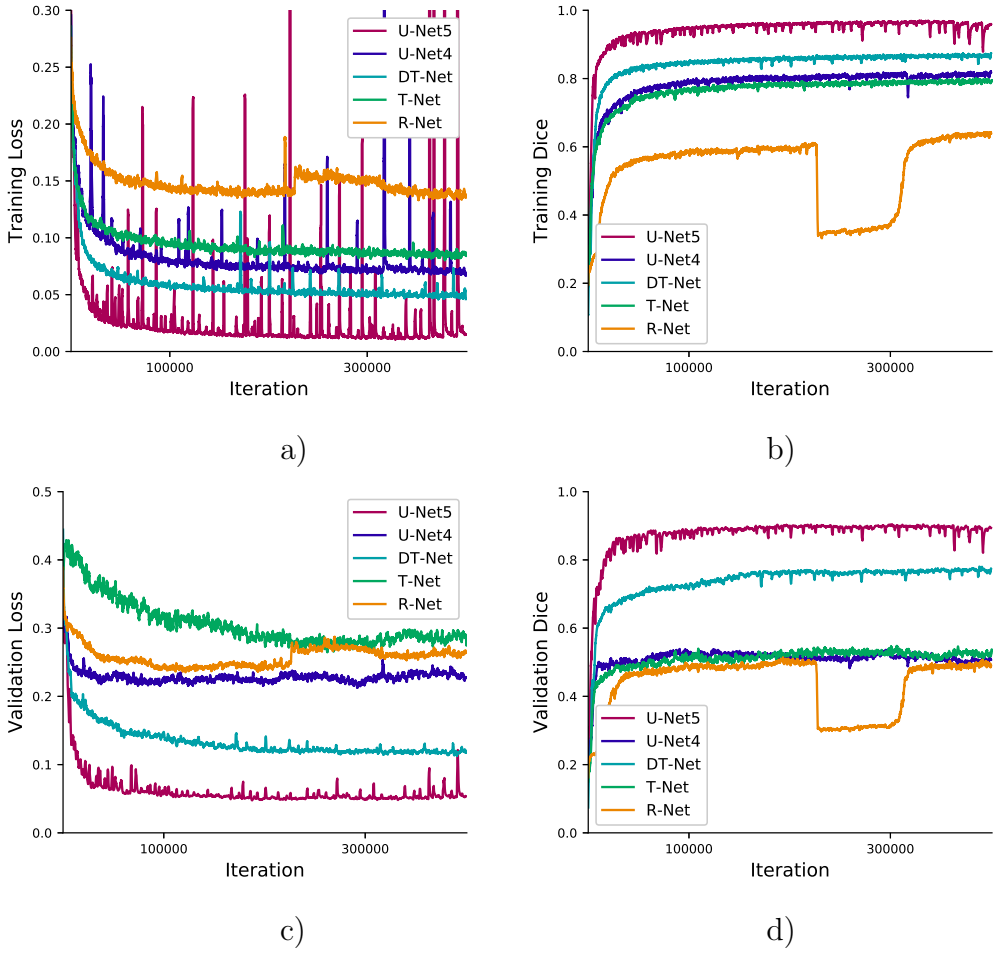


Figure 3.7 Development of the loss (left column) and the Dice coefficient (right column) on the training and validation set of the BraTS data-set. The curves have been smoothed with a moving average filter to make them more discernible. Remarkable is that the validation loss doesn't seem to correspond to the validation Dice. The reason is that the loss is a "soft" function which can decrease by a large margin without changes in the classification performance.

3.8 a)-d)) are available, which means that the input image has four channels. The MRI were prepared identically to the stroke data-set (section 4.1), *i.e.* normalised with respect to the intensity value of normal tissue (section 4.2.2), sampled to provide an equal distribution between ill and sane tissue (section 4.1.1) and augmented (section 4.1.2) with adding a constant and random crops.

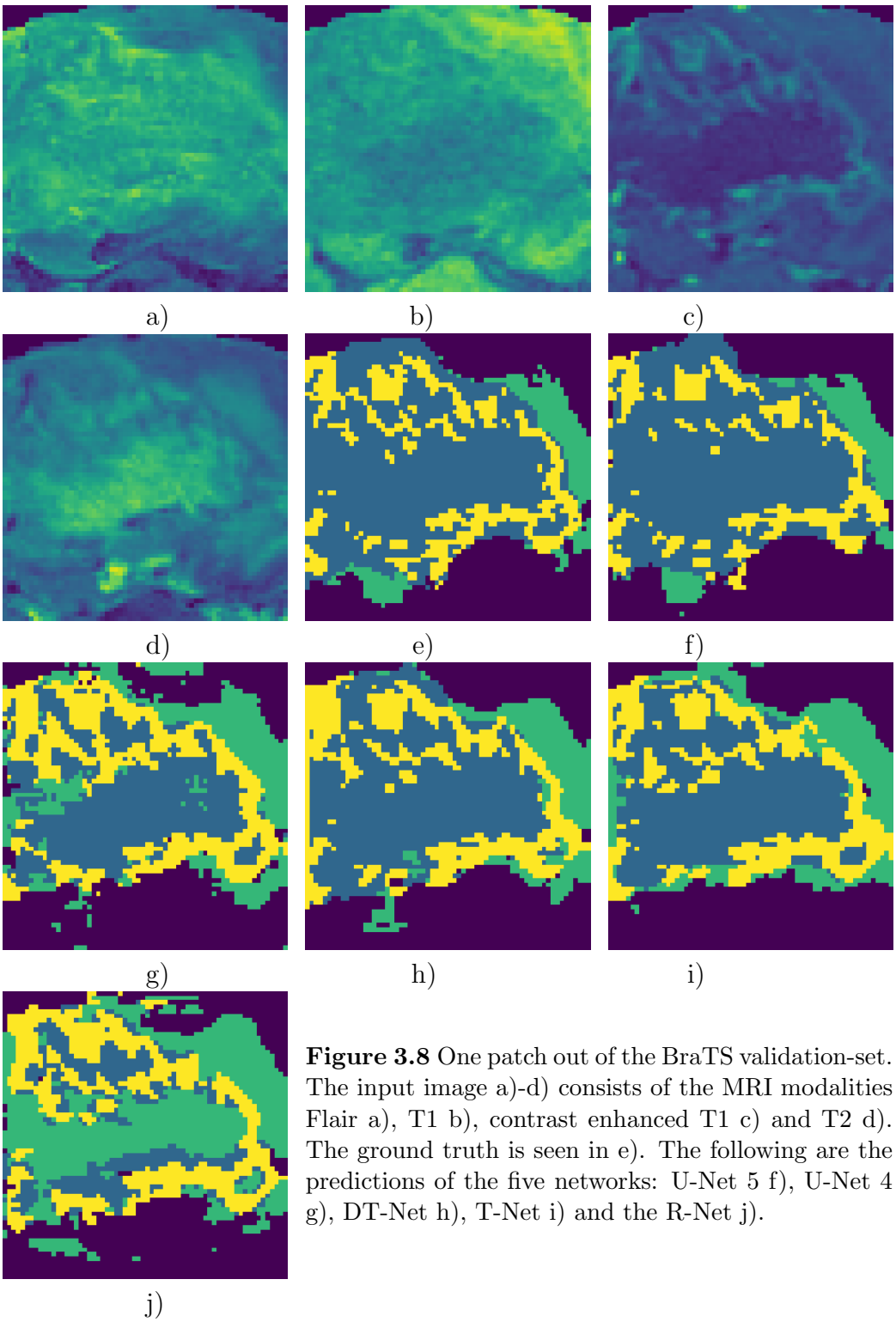


Figure 3.8 One patch out of the BraTS validation-set. The input image a)-d) consists of the MRI modalities Flair a), T1 b), contrast enhanced T1 c) and T2 d). The ground truth is seen in e). The following are the predictions of the five networks: U-Net 5 f), U-Net 4 g), DT-Net h), T-Net i) and the R-Net j).

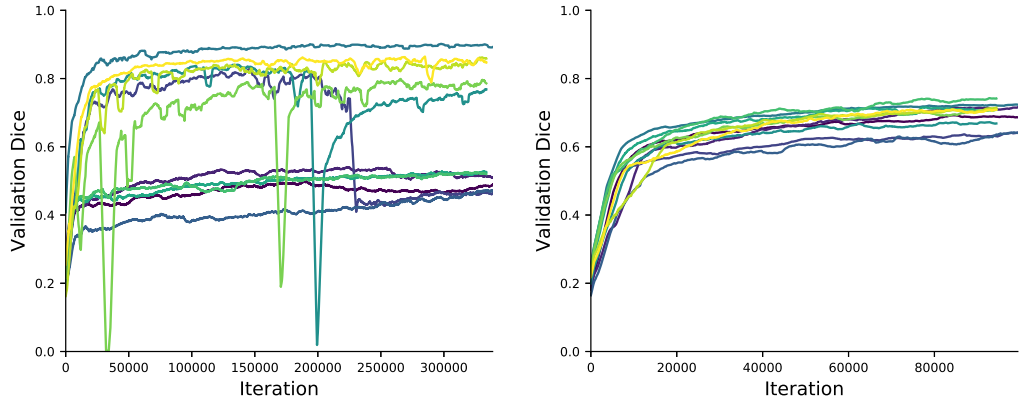
Table 3.2 Network configurations for the BraTS experiment and their final Dice score on the training and validation set.

Model	U-Net 5	U-Net 4	DT-Net	T-Net	R-Net
Kernel Size	3	3	5	5	5
Parameters	31032516	7698116	141929	19359	19359
Learning Rate	0.001	0.002	0.0005	0.002	0.002
m	-	-	10	10	-
Training Speed τ	10737	15032	13322	28761	48767
Training Dice	0.96	0.82	0.87	0.80	0.64
Validation Dice	0.89	0.51	0.77	0.51	0.49

Training All five networks are trained in parallel on this data set, which means that each network receives the identical batches in the same order during training. The networks are configured as in table 3.2 and were trained on an Nvidia Tesla V100 GPU. There was no explicit stopping criterion for the training and the training was continued long after changes in the loss were visible in order to test if over-fitting would appear. The training curves from the run with the highest overall Dice scores are presented in figure 3.7.

Convergence During the training of these models the U-Net would converge to very different solutions in the same training configuration, often not reaching its potential. The convergence of the U-Net and the DT-Net are compared in figure 3.9. In contrast to the transfer block based networks, the U-Net seems to be inherently unstable. The figure shows only the validation Dice, but the training loss and Dice show the same behaviour. This could be an effect of the network size, but it may be interesting to investigate this further.

Training Speed Each network architecture has a different impact on the training time. To asses this the training speed τ was measured in patches per minute (see table 3.2). Please note that the test implementation of the transfer block is a naive implementation of the formulas and Pytorch's maximum pooling is not very efficient for large window sizes. The pooling operation with the largest window size of 129×129 is equivalent to creating a new image and filling it with the maximum of the input image. But implementing this as the latter is 216 times faster than the pooling implementation. So the training speed of the transfer block can be much faster with an intelligent implementation.



a) b)

Figure 3.9 Validation Dice for multiple runs for the U-Net 5 a) and the DT-Net b). The network configurations are identical between runs. The U-Net is highly unstable, different models trained in the same configuration converge to largely different solutions. With a bit of bad luck a well performing U-Net wouldn't have shown up in these experiments and the conclusion would be that the U-Net 5 is outperformed by the DT-Net. This actually happened in the first few runs. The DT-Net however always converges to similar solutions and is therefore more reliable.

Results This real data-set is a way more difficult task than segmenting circles, which allows to see a real difference in the capabilities of the different network architectures. Looking at the loss values the networks are well separated (figure 3.7). But a look at the Dice score shows a lower loss value does not necessarily indicate a better segmentation performance. The networks based on the transfer block perform well. Of course it is not the expectation to beat the U-Net 5 with a network which has only 0.44% of the parameters. But the DT-net comes remarkably close in terms of performance and clearly beats the U-Net 4 which still has 56 times more parameters. Even the T-Net beats the U-Net 4 on the validation set while having less than 20 thousand parameters compared to more than 7 million of the U-Net 4.

This proves that the transfer block works, and that it delivers performance at an astonishingly low parameter cost. Segmentations of one patch are given in figure 3.8. Here the quality of the segmentation of the DT-Net and T-Net is visible. They are very close to the ground truth and the U-Net 5 segmentation while the U-Net 4 segmentation is clearly worse.

3.2 Logic LSTM

All segmentation architectures based on CLSTM use multiple CLSTMs (see section 2.5.5). This is due to inherent problems in the CLSTM itself. Its memory c_t allows it to store information from the past, but for the purpose of prediction it stays a single convolution layer. That means a single CLSTM has the same problems as a single convolutional layer, *i.e.* a small receptive field (see section 2.4.3. The authors of CLSTM based segmentation methods [97, 94, 96, 95] solved this problem with the same approach as for CNNs by stacking layers. Of course this blows up the number of parameters in the models.

A new more elegant approach solves the receptive field problems of the CLSTM and leads to the logic LSTM. The latter is built on two new concepts, the double pass and the logic block.

3.2.1 Double Pass

Developed from the idea of the delayed input (see section 2.5.4) the double pass allows to incorporate the future as well the past of a finite known signal. Basically the model sees the signal twice with a delay equal to the signal length.

First the entire signal is fed to the model, the memory pass, but the model's output is ignored. So all that rests from the memory pass is what the model stores in its memory. Then the signal is fed a second time to the model, the prediction pass. This time the model's output is recovered as prediction. This is visualised in figure 3.10. So for each time step (or slice in the case of MRI) in the signal the model produces a prediction using the current data item plus its memory. The memory contains information about the whole signal, *i.e.* past and future. Furthermore between the time step t in the memory pass and the time step t in the prediction pass exactly the signal length number of steps n passes. This allows the information of step t to be processed for n steps and as a result the prediction from a double pass model is from a deep network. Whereas in the bi-directional approach (section 2.5.4) the prediction for step t is immediate and thus from a shallow network. Also the bi-directional approach uses two distinct models where the double pass uses only one and thus saves 50% of the parameters. It will become apparent in the tests below that the double pass is indeed superior to a bi-directional approach. So compared to the bi-directional approach the double

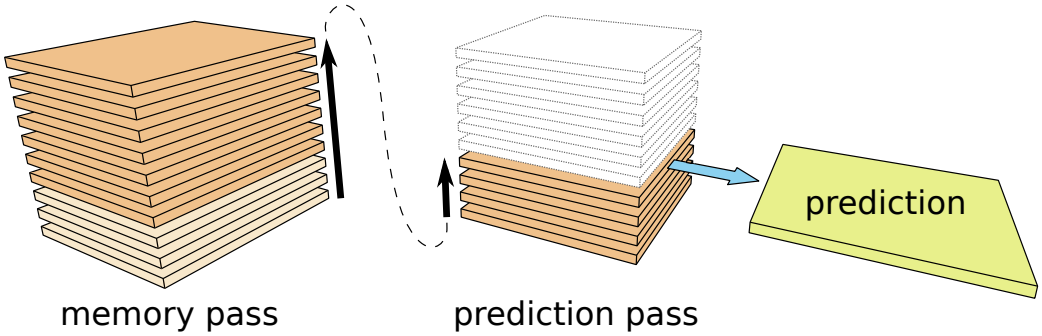


Figure 3.10 The double pass technique displayed at the example of a 3D volume. The recurrent network reads the slices of the volume twice. In the memory pass (left side) it registers important features to its internal memory. In the prediction pass (middle) it produces a prediction for each slice. The steps from the first time it sees a slice and the moment it provides a prediction (indicated by darker colour) allow the network to process the information in the slice over many steps. This recurrent processing results in the prediction (yellow) being from a deep network. In the bi-directional approach (figure 2.7) the prediction is done on the first encounter of a slice and thus from a shallow network.

pass provides better information processing at the identical computational cost while using only half of the parameters.

3.2.2 Logic Block

The logic block builds heavily on the insights of the transfer block. It is designed to replace the convolution layers of the CLSTM (section 2.5.5) and uses a transfer layer to increase the receptive field. As it is an extension of the CLSTM its notation is used here.

For the logic block a partitioning of h_t and c_t is defined:

$$c_t = c_c \,||^4 c_l \tag{3.2}$$
$$h_t = h_c||^4 h_l \tag{3.3}$$
$$r = c_c \,||^4 h_c||^4 x_t \tag{3.4}$$
$$l = c_l \,||^4 h_l \tag{3.5}$$
$$A = c_t||^4 h_t||^4 x_t \tag{3.6}$$

I.e. the hidden state h_t is split into a convolution part h_c with n_{4_c} channels and a logic part with n_{4_l} channels and likewise for the cell state c_t . With

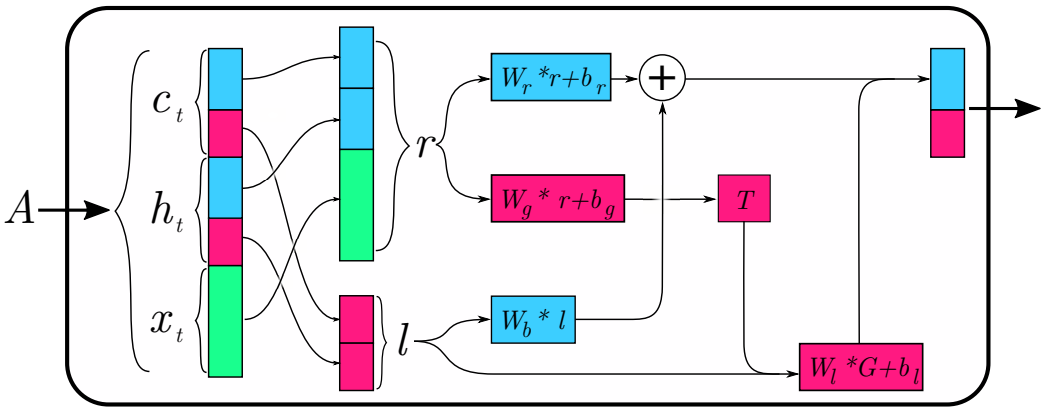


Figure 3.11 Visualisation of the data-flow in the logic block. The colours correspond to the channel numbers. Green = n_{4_x}, red = n_{4_l} and blue = n_{4_c}.

this re-partitioning the logic block is defined as:

$$r' = W_r * r + W_b * l + b_r \tag{3.7}$$
$$G = T(W_g * r + b_g)||^4 l \tag{3.8}$$
$$l' = W_l * (G) + b_l \tag{3.9}$$
$$\mathcal{L}(A) = r'||^4 l' \tag{3.10}$$

The convolutions W_r and W_g are of size $w \times w$, where as W_l and W_b are 1×1 convolutions and T is the transfer layer described above (equation 3.1).

What happens in the logic block is easier to see in figure 3.11. The convolutional parts c_c and h_c are concatenated with the input. The upper blue branch is the convolutional layer known from the CLSTM, it just receives an additional bias from the logic path (equation 3.7). The red path parting in the middle corresponds to a transfer block without the final convolution layer (equation 3.8), instead it is merged with the logic parts c_l and h_l through a 1×1 convolution.

So there is the convolutional branch which works like the original CLSTM (figure 3.11 upper line), working mostly on texture. Then there is the logic branch which exclusively uses 1×1 convolutions (figure 3.11 lower line). This part is like a voxel wise fully connected layer and is able to make decisions for a single voxel, without being distracted from input of neighbouring voxels. The logic branch still can take into account the global situation through the input from the transfer layer (figure 3.11 middle line). In the theory part of the transfer block (section 3.1) it was said that the transfer layer needs to

be followed by a $w \times w$ convolutional layer with $w > 1$ in order to negate the orientation loss through the maximum pooling. As the logic block is designed for a recurrent network, the convolution layer before the transfer layer adopts this job. The 1×1 convolution after the transfer layer is here favourable to ensure clean voxel wise decisions.

Comparing the logic block to a convolution of a CLSTM with $n_{4_{\text{hidden}}} = n_{4_c} + n_{4_l}$ it turns out that the logic block has 33% to 40% less parameters than the convolution kernel, depending on the kernel size w.

In order to calculate this assume identical window sizes of $w \times w$ for the convolutions W_f, W_i, W_o, W_d, W_r, W_g and that the number of channels of the hidden and cell state of the convolutional LSTM $n_{4_{hidden}}$ equals $n_{4_c} + n_{4_l}$, in order to have the same number of channels for both models. Denote the number of channels of the input as n_{4_x}. Then the number of parameters for the simple convolution N_c and the logic block N_l is:

$$
\begin{aligned}
N_c &= w \cdot w \cdot (2n_{4_l} + 2n_{4_c} + n_{4_x}) \cdot (n_{4_c} + n_{4_l}) + (n_{4_c} + n_{4_l}) \\
N_l &= w \cdot w \cdot (2n_{4_c} + n_{4_x}) \cdot n_{4_c} + n_{4_c} && \text{corresponds to } W_r * r + b_r \\
&\quad + w \cdot w \cdot (2c_c + n_{4_x}) n_{4_l} + n_{4_l} && \text{corresponds to } W_g * r + b_g \\
&\quad + 1 \cdot 1 \cdot 2c_l \cdot n_{4_c} && \text{corresponds to } W_b * l \\
&\quad + 1 \cdot 1 \cdot 4c_l \cdot n_{4_l} + n_{4_l} && \text{corresponds to } W_l * (G) + b_l
\end{aligned}
$$

To simplify the calculus assume that $n_{4_c} = n_{4_l} = n_{4_x} = a$:

$$N_c = 10w^2a^2 + 2a \tag{3.11}$$

$$N_l = (6w^2 + 6)a^2 + 3a \tag{3.12}$$

$$\lim_{w \to \infty} \frac{N_l}{N_c} = 0.6 \tag{3.13}$$

For small convolutions kernels like 3×3 the logic block has roughly 67% (at $a = 10$) of the parameters of a comparable convolution in the convolutional LSTM. With taking the limit $w \to \infty$ this descends to 60% which is a clear advantage for the logic block.

3.2.3 Logic LSTM

The logic block $\mathcal{L}(A)$ (equation 3.10) replaces the convolution in the CLSTM and thus defines the logic LSTM:

$$A = c_t||^4 h_t||^4 x_t \tag{3.14}$$
$$f_t = \sigma(\mathcal{L}_f(A)) \tag{3.15}$$
$$i_t = \sigma(\mathcal{L}_i(A)) \tag{3.16}$$
$$o_t = \sigma(\mathcal{L}_o(A)) \tag{3.17}$$
$$d_t = \tanh(\mathcal{L}_d(A)) \tag{3.18}$$
$$c_{t+1} = f_t \circ c_t + i_t \circ d_t \tag{3.19}$$
$$h_{t+1} = o_t \circ \tanh(c_{t+1}) \tag{3.20}$$

Each gate has of course its own weights, indicated by the subscripts to $\mathcal{L}(A)$. The working principle of the logic LSTM is the same as for the CLSTM. It writes to its memory and then reads from it to make a prediction. But the convolution operations have been replaced by logic blocks. Their transfer function provides the logic blocks with a receptive field which is as large as the input. Thus the use of the logic block eliminates the problem of the low receptive field of the CLSTM. Thus a single logic LSTM can do the same job for which an entire stack of CLSTMs was necessary.

Thus the logic LSTM can be wrapped in a simplistic architecture. It is preceded by a single convolution layer which increases the number of channels to $n_{4_{\text{Input}}}$. This allows to control the number of input channels to the logic LSTM and thus gives a fine grained control over the models size. Also this first convolutional layer already extracts features which proved to make the training easier. The logic LSTM is followed by a 1×1 convolution and a softmax layer to provide the final prediction, as recommended by [88]. The hidden state c_t and the cell state c_t are initialised with zeros. The model is trained with the double pass approach (section 3.2.1) and cross entropy is used as loss function (section 2.2.1).

3.2.4 Experimental Validation

With three experiments on synthetic data sets the massive improvements of the Logic LSTM over the convolutional long-short term memory (CLSTM)

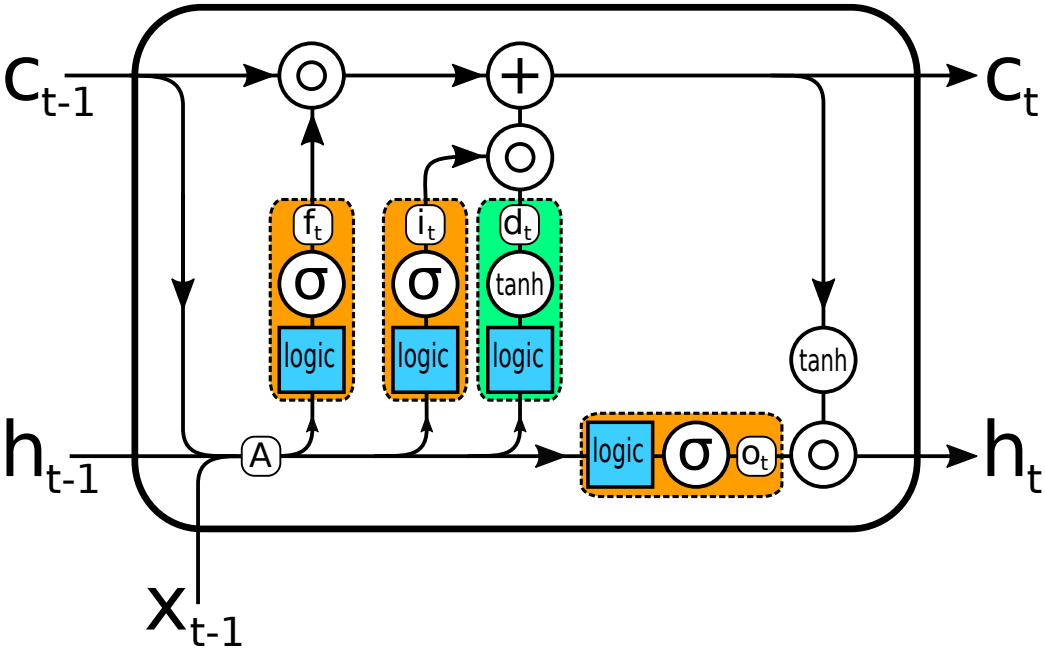


Figure 3.12 The logic LSTM looks very similar to the CLSTM (figure 2.8). The convolution has been replaced by the logic block (blue box).

are shown. Furthermore a 3D U-net [102] and the CLSTM based method of [96] are included to provide a thorough comparison with established segmentation methods. It turns out that the logic long-short time memory (logic LSTM) provides significant advantages over these as well.

Three different data-sets are generated, each one designed to test a specific capability of the models. All are segmentation tasks on binary volumes of size $32 \times 32 \times 24$ and have a massive class imbalance, *i.e.* at least 98% of the volume's voxels are background.

Different Heights This experiment tests the model's ability to measure along the i_3-axis. For a medical data-set this corresponds to remembering information across slices or across time if the input is a video. Two bars of 4 voxels width are randomly placed in the volume, parallel either to the i_1 or i_2 axis. The bars may cross each other at a 90° angle. In the i_1 or i_2 plane the two bars are not distinguishable, but they have different heights in i_3 direction. One is one voxel high and the second 4 voxels. The task is to segment the higher bar.

Different Diameters In contrast to the previous experiment which focused on the i_3-axis, this one is purely about object detection in the i_1 or i_2 plane.

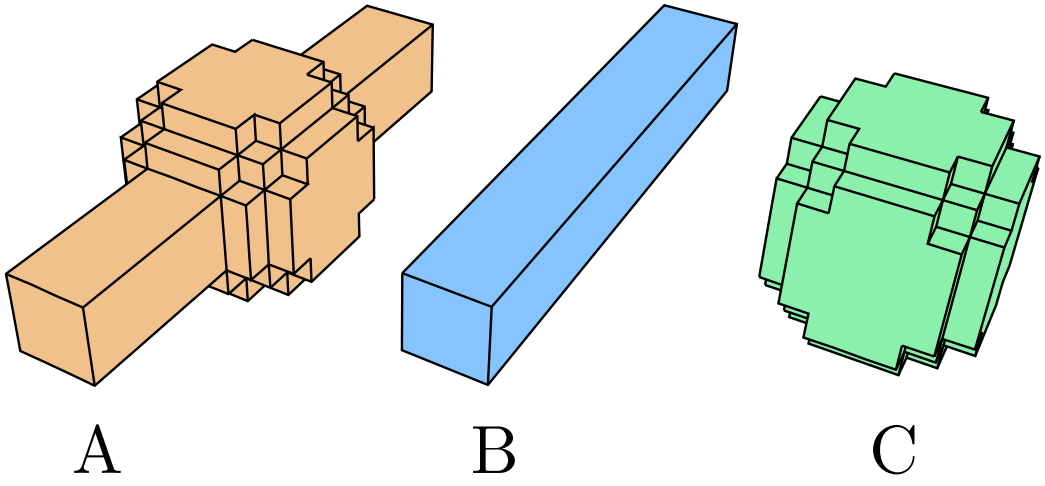


Figure 3.13 The objects used for the simple logic experiment. B is a bar with a 3×3 voxel cross section and C corresponds to a sphere with a radius of 4 voxels. A is the combination of both shapes.

Disks with a radius of 3, 4 and 5 voxels and a height of one voxel are randomly placed but are not allowed to overlap, but they can touch. Through the random distribution the slices contain one to three disks. The task is to segment the disks with radius 4. What makes this task difficult is not only the need to differentiate between very similar objects but also that there is no correlation along the i_3-axis.

Simple Logic The last two experiments can be solved with local decisions. But the last experiment tests if the model is able to make global decisions, taking into account the whole volume. Three 3D objects A, B and C (see figure 3.13) are placed randomly without overlap. Object A is always placed. Object B is placed with 50% probability and Object C is placed with 70% probability. The task is to segment object A if and only if object C is present in the volume. Object B is similar to object A and serves only as a confounding object. Three things are tested in this experiment: if the model can distinguish separate objects, if it can recognise objects by their shape and if it can implement a logical AND operator.

A standard CLSTM cannot solve these tasks as it has no access to future slices. For example in the first experiment it cannot know if a bar will continue on the next slice, *i.e.* is to segment, or not. Thus a double pass CLSTM (DP-CLSTM) and a bidirectional CLSTM (Bi-CLSTM) are included in place. This also allows to verify that the double pass is a valid replacement for a bidirectional approach.

The following five models participate in the experiments. All models are followed by a softmax layer to produce the probability maps for the classes and trained using cross-entropy with a learning rate of 0.001.

DP-CLSTM The CLSTM [93] trained with the double pass approach as described above. Convolution kernels are of size 5×5 and the hidden and cell state have $n_{4_{\text{hidden}}} = 10$ channels. A 3×3 channel increasing convolution layer increases the channel number to 8 before the CLSTM cell. The output of the CLSTM is passed through a 1×1 convolution layer which reduces the number of channels to 2.

Bi-CLSTM Two CLSTMs are trained in a bidirectional manner [92]. Their output is concatenated along the channel axis i_4. It is wrapped with the same channel increasing and reduction layer as the DP-CLSTM.

Logic LSTM The logic LSTM described above with the following configuration: $n_{4_{\text{Input}}} = 8$, $m = 2$ and $n_{4_c} = 2$.

CLSTM-U-Net The architecture by Arbelle et al. [96] is basically a U-Net [87] where the channel increasing layers have been replaced with CLSTMs and with 4 different resolutions instead of five. For our experiments the CLSTMs needed to be replaced by bidirectional CLSTMs. The number of start channels after the first channel increasing layer is set to 8.

3D U-Net The 3D version of the U-Net features batch normalisation layers and also works only with 4 different resolutions [102]. Like all other models the number of start channels after the first channel increasing layer is set to 8.

While the first task was learnt by all five models, the ones using CLSTMs are taking 10 times more iterations to converge than the logic LSTM. For the 3D U-Net this task is almost trivial thanks to its 3D convolution operation which can directly measure the height of the bars.

But what was an advantage becomes detrimental in the second experiment. Here the 3D U-Net struggles because there is no correlation along the i_3 axis and most of the convolution kernels need to be set to zero, effectively making them 2D convolutions. It learns the task but does not achieve a perfect segmentation, always misclassifying some stray pixels. The CLSTM-U-Net performs on the second task almost like on the first one.

The real revelation for the second task are the results of the Bi and DP CLSTM. As the context along the i_3 axis is useless the Bi-CLSTM behaves like a single convolution layer. Its 5×5 convolution kernel is too small to capture the entire circles and as a consequence it cannot differentiate between

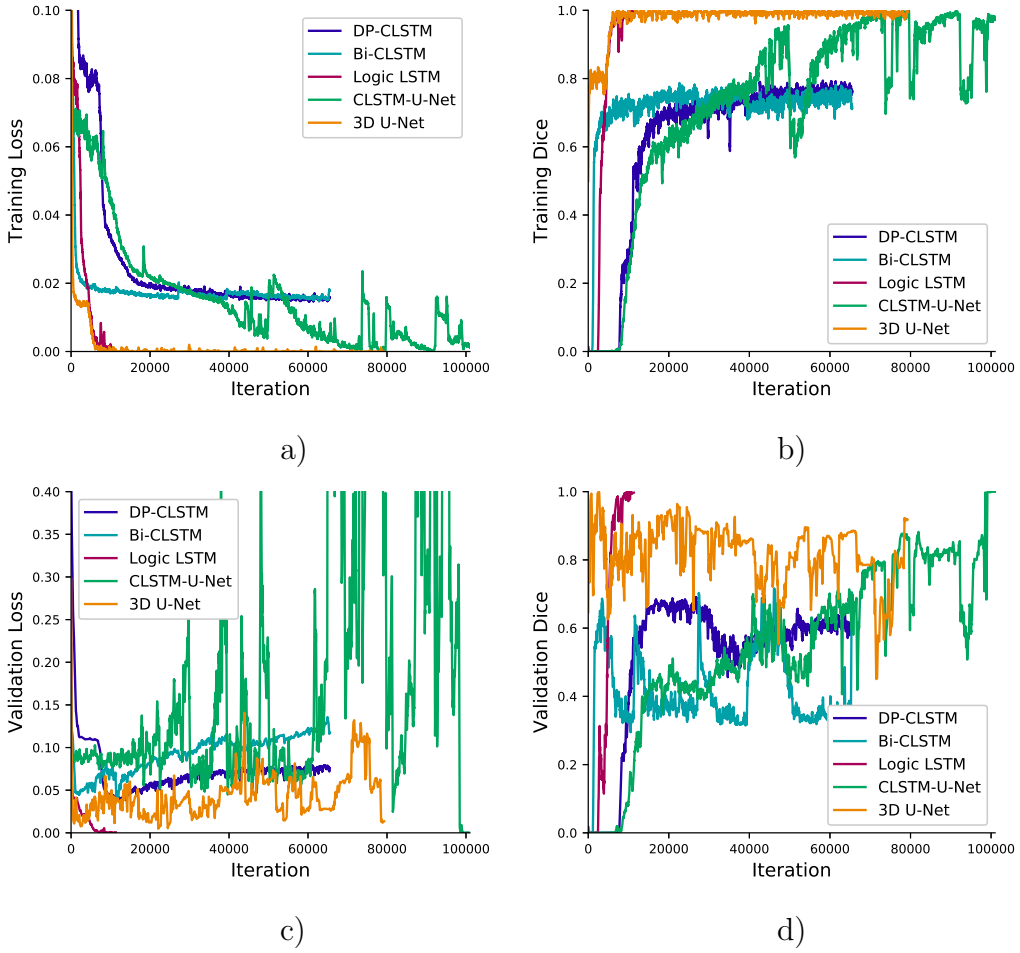


Figure 3.14 Development of the loss (left column) and Dice coefficient (right column) on the training set (first row) and the validation set (second row) for the "simple logic" experiment. Our method, the logic LSTM (red), converges very quickly and is thus only visible on the very left side of the plots. The Bi and DP-CLSTM show clearly a classic case of over-learning. Finally the CLSTM-U-Net demonstrates that the validation error and the segmentation performance are not necessarily related.

the circles of radius 4 and 5 and fails to learn the task. The DP-CLSTM however delivers a much better result. Even though the segmentation was not perfect, few random voxels were wrongly classified, the correct circles were segmented. It sees the circles already in the memory pass, and when it sees them again in the prediction pass it can use information from its memory c_t. From the memory pass to the actual segmentation 24 images are passed through the network, which makes it a network of depth 24. Apparently this

Table 3.3 Summary of the trials on the synthetic data-sets. It includes the number of the models parameters N. For each experiment the success on the task is given, as well as the (approximate) number of iterations to convergence ρ.

Model	Bi-CLSTM	DP-CLSTM	**Logic LSTM**	CLSTM-Unet	3D U-Net
N	56330	28270	**22138**	910386	1193762
Experiment: Different Heights					
Success	Yes	Yes	**Yes**	Yes	Yes
ρ	50 000	50 000	**4 800**	45 000	2 300
Experiment: Different Diameters					
Success	No	Yes	**Yes**	Yes	Yes
ρ	-	80 000	**5 500**	38 000	30 000
Experiment: Simple Logic					
Success	No	No	**Yes**	Yes	No
ρ	-	-	**7 400**	99 000	-

is enough to increase the receptive field to the necessary size for the circles segmentation task. This validates the double pass approach. Not only is it a replacement for the bi-directional approach, it also halves the amount of parameters needed while maintaining the same amount of computations and adds some additional value in the form of a larger receptive field.

The logic LSTM has no trouble learning the second task. This confirms that the transfer layer T increases the receptive field to the whole image, as intended.

In the last experiments both Bi and DP-CLSTM fail due to their inherent problems with the receptive field. Also the 3D U-Net fails. It achieves a perfect result on the training set but is not able to generalise to the validation set. Only the CLSTM-U-Net and the logic LSTM learn the third task and can provide a perfect segmentation on the validation set (figure 3.14). Thus the logic LSTM effectively solves the receptive field problems of the CLSTM while avoiding the use of multiple CLSTM as it is done in other solutions [97, 94, 96, 95]. The last experiment also verifies that the logic block is indeed able to implement basic logic.

The three experiments have shown that the logic LSTM is on par with established segmentation methods and can segment taking the whole input volume into account. Looking at the models parameters a decisive advantage of the

logic LSTM becomes apparent: it uses less than 3% of the parameters of the established methods! On top of that the logic LSTM is easy to scale to arbitrary input sizes without a significant growth in the number of parameters. The 3D U-Net on the other hand expands from 1 Million to 48 Million parameters if a fifth resolution is added, which would be necessary for larger input volumes.

Chapter 4

Stroke MRI Segmentation

The main goal of this work is to find the signs of stroke relevant for diagnosis on MRI using machine learning techniques. The most relevant are the lesion and the thrombus. So this translates to the task of an automatic segmentation of lesion and thrombus from stroke MRI. This chapter describes the data-set and pre-processing steps and the results achieved with different automatic methods.

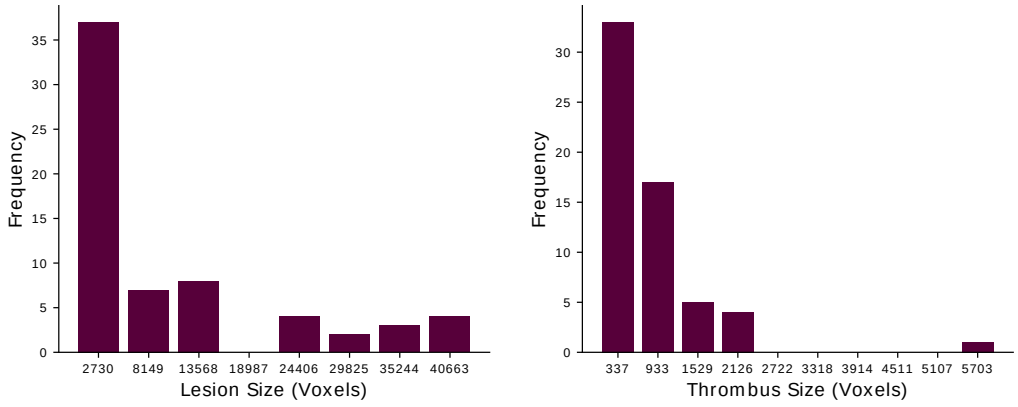
4.1 Stroke Data-Set

The stroke data-set was obtained through the support of the neurology group of the Centre Hospitalier Sud Francilien lead by Prof. Didier Smadja. It contains cranial MRI of 65 stroke patients with a thrombus in the middle cerebral artery. The MRI modalities available for this data-set are DWI B0 ADC FLAIR ToF, SWAN and the corresponding Phase. 61 of the patients show a thrombus on SWAN, but some exams are lacking the Phase image. Thus there are only 58 patients with a thrombus visible on SWAN where all 7 modalities are available. All 65 patients have a visible lesion. The MRIs were acquired from a 1.5 T and a 3 T General Electrics MRI machine. DWI B0 and ADC share the same resolution, as they are from the same acquisition. The other modalities differ with the resolutions given in table 4.1.

The number of slices is different between modalities because of the different slice thickness of the acquisitions. It differs also between patients as the number of slices is adjusted to the height of the patients head. So a scan from a person with a large head has more slices than the scan from a person with a small head. A further complication is that the head's position within the scanned volume is not guaranteed to be the same across scans. In order to

Table 4.1 Sizes of the volumes obtained for the different MRI modalities.

Modality	Slice resolution	Number of slices
DWI/ B0/ ADC	256×256	24 to 38
FLAIR	512×512	24 to 40
ToF	512×512	124 to 184
SWAN	512×512	72 to 216



a) Lesion b) Thrombus

Figure 4.1 The size distribution of the lesions a) and thrombi b) in the CHSF stroke data-set. Most of the lesions are small which confirms that the MRI are from the hyper-acute phase of stroke. For the purpose of this figure the lesion or thrombus was calculated as the union of the multiple segmentations available for each patient.

spatially link the modalities a co-registration is necessary (see section 4.2.1).

For both the lesion and the thrombus manual segmentations are available. Each patient has been segmented by at least two neurologists. The inter-observer agreement, measured in terms of the Dice coefficient (section 2.2.1), was 0.69 ± 0.15 for the lesion and 0.61 ± 0.18 for the thrombus. This is a result which is comparable to the inter-observer agreement of the ISLES 2015 stroke lesion data-set [64] of 0.7. Furthermore a Dice of 0.7 is close to human capacities for objects of this size. As no protocol was established on how to segment exactly and none of the neurologists had ever before done a manual segmentation, this is an astonishingly good result and shows the high expertise of the neurology group of the CHSF.

The distributions of the lesion size (figure 4.1 a)) clearly shows the predominantly small lesion sizes in the data set with a median size of 2920 voxels. This is due to the MRI being taken in the hyper-acute phase of stroke where the lesions are still developing and have not reached their final size. The distribution of the thrombus sizes (figure 4.1 b)) reminds of a Poisson distribution, which is not surprising considering that the thrombus size can be considered a random event due to the many possible causes for a thrombus. It also shows that the thrombi are rather small objects, with a median size of 542 voxels which makes it difficult to find them.

4.1.1 Sampling

Sampling in this context is the creation of sub-images from an MRI for the use in a machine learning method. This is necessary because the whole MRI of a patient is too large to be processed in one piece. This in turn means that the machine learning models don't see the whole MRI. The sub-images, called patches, have to show relevant parts of the brain in order to produce high-performance models. For example a thrombus has an median size of 542 voxels. If patches are randomly sampled from a SWAN with roughly 40 million voxels the chance is high that the thrombus never appears on a single patch. Because of this class imbalance the models trained on these patches can never learn what a thrombus looks like. Other problems which arise are border problems, where thrombi situated at the edges of a patch have not enough context to be correctly identified. Then there are also objects which look very much like the thrombus and are likely to cause false positives. These "likely false positive" objects are also rare with respect to the millions of voxels in an MRI.

To counter all these problems a specific sampling strategy has been chosen which ensures that the training sets contain relevant data. Border problems are avoided by an overlapping patches approach as proposed by Ronneberger et al. [87]. That means that the border of a patch is ignored and only the predictions of the centre, where enough context is available, are used. As the predictions are only for the centre of the patches it is safe to create the patches in a way that the relevant information is centred. The class imbalance is eliminated by choosing an equal number of patches with thrombus and "likely false positive" objects. During training it is assured that every second image in a mini-batch shows a thrombus and every other shows a "likely false positive" object. This forces the models to concentrate on the differences between thrombus and "likely false positive" objects. Easier to learn non-

thrombus tissues are learned alongside as non-thrombus tissues are always on the patches surrounding the thrombus. The patch size has to be chosen small enough that a thrombus makes up a relevant amount of voxels in the patch (at least 1%) and large enough that the surroundings are visible. "Likely false positive" objects are defined as areas with the same intensity range as the thrombus inside the brain. This sampling method applies to the lesion as well.

4.1.2 Data Augmentation

Data augmentation is the process of increasing the size of a training set by adding altered versions of the already existing samples. It is used when the training set is too small to generate a model which generalises well. Also the CHSF stroke data-set is small as it has only 65 patients.

The data augmentation step is part of the training schedule. During training each patch is "augmented" (= transformed) before being used. If the patch is reused, it is augmented differently. This means the method never sees the original patches from the sampling step (section 4.1.1). As the augmentation includes random steps this also means that the training set is of infinite size and never repeats. This allows learning decent models from small data-sets and almost eliminates over-fitting.

For the stroke data set random crops and adding of a constant were used as data augmentation steps. Random crops in connection with the sampling of patches works as follows: The patches are sampled at a size which is larger than the size intended for training. For example the model is trained on 64×64 then the patches are sampled at 74×74. Then during training a 64×64 image is cut from the 74×74 patch at a random location but such that the 64×64 image is entirely contained in the 74×74 patch. The number of possible random crops for a patch is determined by the size difference of the patch and training image size.

4.2 Pre-Processing

The MRIs are pre-processed in various steps before they can be used for training a model. Denoising is not part of the pre-processing pipeline because the MRI are denoised during acquisition. The remaining noise is not of a relevant magnitude. But various normalisation steps are indispensable,

notably intensity normalisation and spatial alignment.

4.2.1 Co-Registration and Data Format

All the MRI modalities have different resolutions. As the patient may move his head between scans the head position and inclination may be different as well. To bring all images to the same resolution the co-registration algorithm of the SPM 12 toolbox[1] for Matlab is used. The co-registration target is the SWAN, as it has the highest resolution. A down-sampling of the SWAN may loose the thrombus, compared to this a quality loss from up-sampling and rotating the other images is negligible.

Another use for the SPM 12 package is the data format conversion. The MRIs extracted from the picture archiving system (PACS) of the CHSF are in DICOM format. This format is very inconvenient to work with as it is basically a container for all sorts of image and other data. It is transformed to the Nifti file format which offers file compression and is generally easier to load and write than DICOM files. In the case of the CHSF DICOM files the MRIs are stored as an unordered collection of slices, which makes it necessary to figure out which slice belongs to which modality and which are the previous and following slices. The header which contains the order of the slices is in some unreadable non standard format. The SPM 12 package did a good job at sorting out the different modalities but for some patients the DWI and B0 were exchanged and had to be manually corrected.

4.2.2 Normalisation

MRI is a qualitative and no quantitative measurement [103], *i.e.* the intensity values and range of repeated measurements of the same patient are different even though it appears to be the same image to human eyes. Across patients this variability is even higher (compare figures 4.2 a) and c)). Therefore a normalisation of the images is necessary to obtain repeatable results with automatic processing of the images.

Luckily the MRIs all have a common characteristic. Most of the brain is composed of healthy tissue which is visible as a large peak in the histogram of the voxel intensities, as indicated by the red lines in figure 4.2. This peak is only overshadowed by the peak corresponding to the empty space around

[1] http://www.fil.ion.ucl.ac.uk/spm/software/spm12/

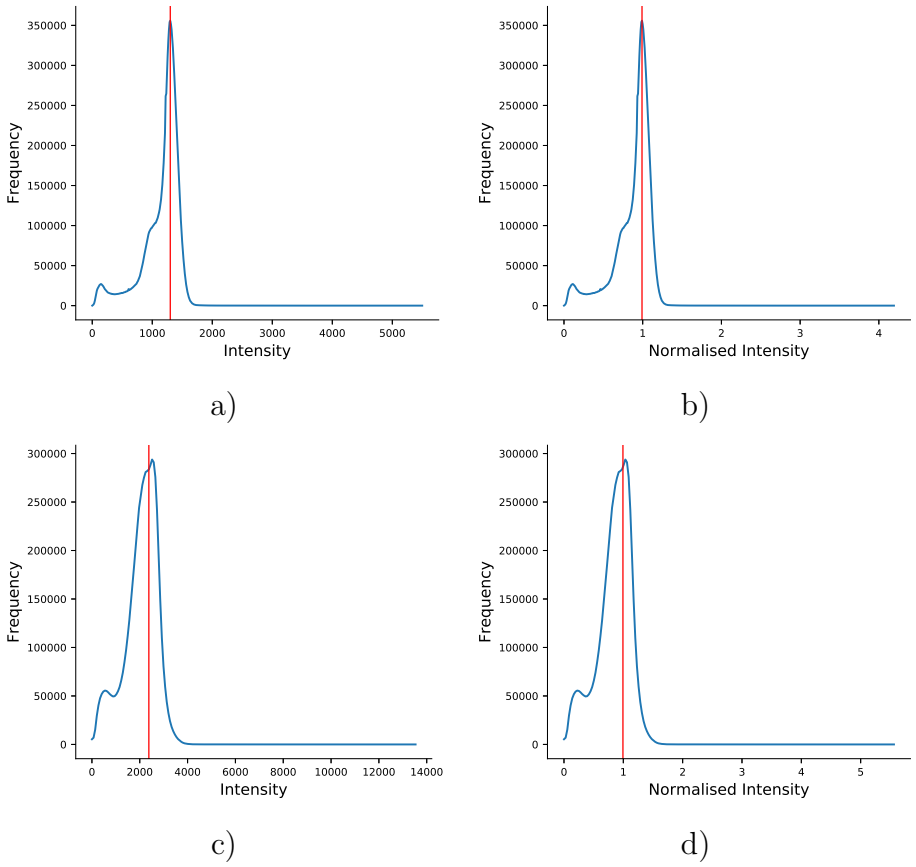


Figure 4.2 Two examples of the intensity normalisation of the SWAN histogram. Note the location of the peak corresponding to the healthy brain tissue (red line) in the original histograms a) and c). Both histograms show this peak, but its location is different. This illustrates why intensity normalisation of MRI is important. Then the intensity values are divided by the peak intensity value and the normalised histograms b) and d) are obtained. Note that this histogram shows intensity values of the brain only, the very large peak around zero of the empty space around the head has been removed for readability. The histograms of other MRI modalities are very similar and the normalisation process is identical.

the head. But with the brain mask from the skull stripping step (section 4.2.3) this peak can be removed easily and the peak of the healthy tissue is the dominant peak in the histogram of the brain. The peak's centre is not always the highest point, as seen in figure 4.2 c). Therefore the peak's centre is determined by fitting a mixture of two Gaussians to the histogram. The MRI is normalised by dividing it by the peak value, *i.e.* setting the value of the peak to 1. Thus on the normalised images the most obvious

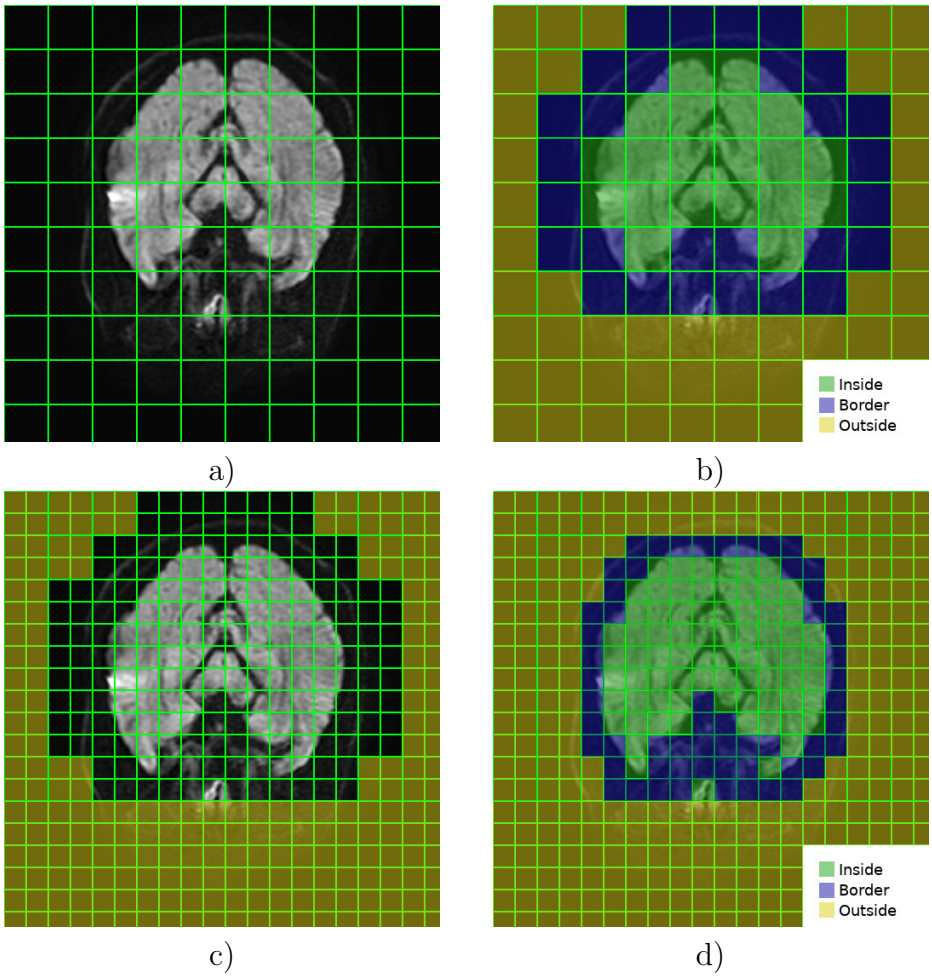


Figure 4.3 Two iterations of the skull stripping algorithm. The image is divided into 3D cubes a). Then the cubes are classified into inside and outside according to the mean intensity of the cubes. The outside cubes adjacent to inside cubes are labelled as border b). Outside cubes are excluded from further calculations and inside and border are divided into smaller cubes c). Then the smaller cubes are once again classified into inside outside and border d). This process is repeated until the smallest cube size is reached.

reference point, the healthy tissue, has always the same value. Should the centre estimated with the mixture of Gaussians be slightly incorrect this has no effect because of the small added constant from the data augmentation step (section 4.1.2) where it is shifted anyway. This normalisation method is a simplified version of the normalisation by histogram landmarks of Nyul et al. [104]. Throughout this work only normalised MRIs are used, with two exceptions.

The ADC measures an approximation of the diffusion coefficient and the phase of the SWAN measures the phase of the RF signal. Both are quantitative measurements of physical quantities without need for a normalisation.

4.2.3 Skull Stripping

Skull stripping is the process of extracting the brain from an MRI image. In most MRI sequences only the skull is visible amongst the non brain tissues hence the name. The process is very important for two reasons: reduction of the data size and anonymisation. The MRI contain the facial bones which would allow the reconstruction of the face. Removing these bones renders the MRI anonymous. Then the brain makes up less than one quarter of the MRI volume and knowing where it is reduces the computational load significantly.

Skull stripping is a well known task and a good overview is given by [46]. Unfortunately this vast selection of methods is not applicable to the CHSF stroke data set. All skull stripping methods use T1 or T2 weighted images and the stroke data-set has neither. On the other hand the brain can be almost perfectly segmented on DWI. There are only some bone artefacts and a few other points with high intensity.

Upon this observation a new skull stripping algorithm for DWI was developed. The singular high intensity points outside the brain are too large to be eliminated with a median filter and they may be connected through fine bridges with the brain so a thresholding of the DWI combination with a connected component analysis doesn't work as well. The solution is a multi-resolution algorithm which is able to eliminate small objects outside the brain and can still segment the brain with high precision.

The algorithm is:

Input A volume V where objects to segment are bright. A start number of cubes k. A threshold τ, which can be set manually. Here it is calculated as $\tau = \text{mean}(V) + \text{std}(V)$. With $\text{std}(V)$ being the standard deviation of the intensity values in V.

Do Divide V into k^3 non overlapping cubes of size $\lceil n_1/k \rceil \times \lceil n_2/k \rceil \times \lceil n_3/k \rceil$ and store them in a list of active cubes L. If the dimensions n_i of V are not evenly divisible by k some cubes at the border of the V will be of smaller size. As the border of the image doesn't contain brain this is irrelevant, they will be eliminated in the first iteration.

While The axis sizes of all cubes in L are > 1

Do All cubes C in L are labelled as "inside" if mean$(C) > \tau$ and "outside" if not. Cubes labelled "outside" which are adjacent to an "inside" Cube are labelled "border.

Do All cubes labelled "outside" are removed from L.

Do Each cube C in L is divided into 8 sub-cubes of equal size. C is removed from L and its sub-cubes are added to L.

Do Construct a new volume B with the same size as V. The voxels corresponding to a cube in L get the value 1, all others get the value 0. The largest component in the binary image B is found with a connected component analysis and labelled as brain. All voxels in B which are not labelled as brain are set to 0.

Return B

The algorithm is visualised in figure 4.3. Small cavities in the brain will be ignored by the algorithm but the ventricles will be labelled as non brain as well. Also larger crevices like the ones between the hemispheres. To produce a classical skull stripped brain these holes need to be closed. This is done with ray tracing shadows along the three main axis of the volume. That makes a total of 6 shadows which are cast. Regions which are covered by at least 4 shadows are certainly inside the brain and are added to the brain label.

After the first step this algorithm treats only the relevant regions of the volume, which makes it very efficient. It can be run in a couple of seconds and therefore it is well adapted to be used in a real clinical scenario. This also sets it apart from other skull stripping techniques which often calculate complicated contours and are rather slow.

4.2.4 Lesion Enhancement

As explained in section1.2.2 the lesion can be seen on DWI but is easily confused with a couple of imaging artefacts. ADC tries to remedy this but has the disadvantage that it is a very crude approximation of the diffusion coefficient. The directional dependency of the diffusion coefficient and local imaging artefacts are disregarded by ADC and can render small lesions invisible (compare figure 4.4 b)). Remember that the ADC is calculated from the logarithm of the MRI signal (i.e. the DWI):

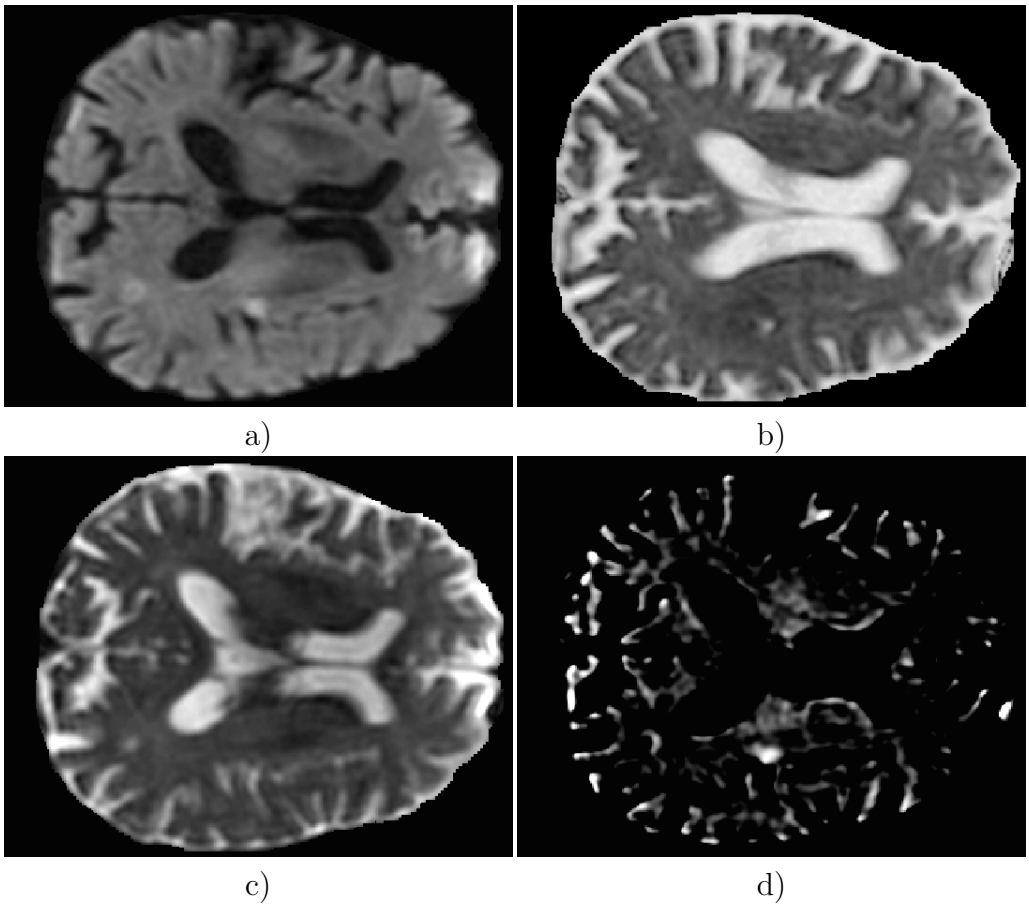


Figure 4.4 Some lesions are very small and barely visible on DWI a). On ADC b) they completely disappear as the assumptions of the ADC calculation are not true. The B0 c) however captures all defects of the magnetic field and can be used to remove most artefacts and increase the lesion visibility. The enhanced image d) clearly shows the lesion and all major artefacts are removed.

$$\ln(S) = -bD \tag{4.1}$$

Where the diffusion coefficient D is approximated as the apparent diffusion coefficient by linear regression from measurements with different magnetic fields b. The linear regression is not able to entirely capture the complexity of the diffusion process.

But the actual interest is in the lesion and not the diffusion coefficient. The lesion is visible on DWI but obstructed by complex interactions of the brain

and the magnetic field. Most of the artefacts seen on DWI are due to magnetic inhomogeneities and unwanted magnetisation of metallic parts of the MRI machine which also alter the magnetic field. The impact of b on the artefacts is negligible because b is really small compared to the other magnetic fields. Luckily there are two measurements from the DWI sequence, the DWI and the B0. The B0 is taken with $b = 0$, so no diffusion effects are seen but most artefacts still appear on this image. So all the complex interactions which are lost on the ADC are directly measured on B0! Then the actual DWI captures the diffusion process with the help of the additional magnetic field b. But unfortunately B0 and DWI are not directly comparable.

In section 4.2.2 the need of a normalisation was discussed. It turns out that this normalisation also makes the B0 and DWI comparable. An enhanced image E can be calculated as

$$E = DWI_{normalised} - B0_{normalised} \tag{4.2}$$

from the normalised DWI and B0. All the magnetic background interactions are simply subtracted from DWI and the interaction related to diffusion remain with high values. It turns out that regions with restrained diffusion have positive values on E and normal regions have negative values. The normalisation has introduced a natural threshold with the value of 0 for indicating lesion regions. The thresholded E (figure 4.4 d)) clearly enhances the lesion and diminishes artefacts compared to DWI. Therefore the enhanced image was included in all supervised lesion segmentation approaches (section 4.4).

4.2.5 Evaluation Metrics

The quality of the results of an automatic segmentation method has to be quantified. For this the segmentation provided by the method is compared to a ground truth, which is the manual segmentation by an expert. The most common evaluation metric for evaluating segmentations of bio-medical images is the **Dice coefficient** (section 2.2.1) which measures the overlap of two segmentations but also takes into account their cardinality. This dependency on cardinality makes it difficult to tell which value of the Dice coefficient indicates a good segmentation result. A perfect segmentation is identical to the ground truth and gives a Dice of 1. If there is no overlap the Dice is 0. Usually a segmentation where the border is displaced only

by one pixel from the ground truth is considered a good segmentation. But for this case the Dice varies depending on the surface or volume of the segmented object. Thus the Dice is not an absolute measurement of quality and a good Dice has to be defined with respect to the inter-observer Dice on the given data-set. For this the same image is segmented by multiple experts and the Dice of the experts' segmentations, the inter-observer Dice, is calculated. For the thrombus segmentation task on the CHSF data-set the inter-observer Dice is 0.61 and for the larger lesion it is 0.69. For even larger objects like lungs good Dice coefficients are above 0.9 [105]. If the Dice coefficient between the automatic segmentation and the ground truth is close to the inter-observer Dice the segmentation is considered to be human level performance, *i.e.* good. While the Dice captures the overlap there are other factors which determine the segmentation quality.

In a classic classification task the number of **false positives** and **false negatives** plays an important role. False positives are voxels which have been segment but are background whereas false negatives are voxels which are not segmented but are part of the ground truth. For the stroke segmentation task this is not a useful metric as these numbers do not correspond to anything a human would perceive, *i.e.* they are not visually related to the ground truth. What a human sees on a segmentation are connected components and if these components correspond to the components of the ground truth. Thus a useful metric for segmentation quality is the number of false positive components and the number of false negative components similar to the definition in [65]. For this a connected component analysis [106] is used to partition the segmentation S into a set of connected components $\mathcal{S}$. Also the ground truth T is partitioned into a set of connected components $\mathcal{T}$. The number of components in the segmentation which have no overlap with the ground truth is the **number of false positive objects (FP)**.

$$f_p = \{s \in \mathcal{S} | s \cap T = \varnothing\} \tag{4.3}$$
$$\mathrm{FP} = |f_p| \tag{4.4}$$

Likewise the number of components in the ground truth which have no overlap with the segmentation is the **number of false negative objects (FN)**.

$$f_n = \{t \in \mathcal{T} | t \cap S = \varnothing\} \tag{4.5}$$
$$\mathrm{FN} = |f_n| \tag{4.6}$$

The importance of these measures is intuitive. If false positive objects are present the segmentation would point the medical doctor to regions of the

brain where there is not thrombus. False negative objects mean the real thrombus has not been found. This information is complemented by the **average size of the false positive objects (FP size)** and the **average size of false negative objects (FN size)**.

$$\text{FP size} = \frac{1}{|f_p|} \sum_{a \in f_p} |a| \tag{4.7}$$

$$\text{FN size} = \frac{1}{|f_n|} \sum_{a \in f_n} |a| \tag{4.8}$$

Together with the size of the objects on the ground truth this helps to determine how important the false positive objects are with respect to the ground truth. For example a lesion could have a size of 10 000 voxels. If there are false positive objects in the segmentation with a size of 30 voxels they are just small spots which would not disturb diagnosis. However if the false positive objects are within the size range of the actual objects which should be segmented the segmentation has to be considered bad, even if the dice coefficient might not indicate this.

Of course the segmentation performance is not measured on one segmentation alone but on an entire test-set. Thus the FP, FP size, FN and FN size given for the test-sets below are actually the average over the whole test-set. The last metric used in this work is the **detection rate** D_r which has to be defined for a whole test-set. Be a the number of patients in the test-set where $S \cap T \neq \varnothing$ and b the total number of patients in the test-set. Then $D_r = \frac{a}{b}$. This definition can lead to a situation where the detection rate can be 1 and at the same time FN is not zero. The reason is that some patients can have multiple thrombi where only one is detected and the detection rate counts only if something was detected. Nevertheless the detection rate is still a valid measure because on the one hand multiple thrombi are rare and on the other hand the secondary thrombi usually don't influence the treatment decision.

4.3 Automatic Thrombus Segmentation

Segmentation of the thrombus is a very difficult task. It cannot be seen directly on the MRI images. Instead its presence is indicated by a secondary effect. The thrombus is constituted by clotted blood, which contains a lot of densely packed haemoglobin proteins which in turn contain iron atoms. This

high local iron concentration causes a defect in the magnetic field of the MRI. Most MRI sequences compensate for magnetic field defects which renders the thrombus invisible. But on susceptibility weighted imaging sequences this defect is visible as a dark spot. Usual clinical T2* sequences are of too low quality, due to time constraints in a stroke scenario, to see anything but the largest thrombi. Other susceptibility weighted imaging sequences, for example the susceptibility weighted angiography (SWAN), produce higher quality images in the same time where even the magnetic defects of small thrombi are visible.

So physics dictate which MRI modalities should be used to detect the thrombus. These are certainly the SWAN and possibly its phase. The ToF may give additional clues in as some arteries may show up as black on SWAN and thus could be sorted out using the ToF. The other MRI sequences have no value for detecting the thrombus. Preliminary experiments with spectral analysis, classical image processing, *i.e.* texture, filters and similar methods lead to the insight that the thrombus has identical characteristics as any other dark area. Also principal component analysis, independent component analysis, empirical mode decomposition, clustering algorithms and support vector machines could not find the thrombus. Therefore it was clear that the clue to finding the thrombus lies not in the voxel values or local texture but rather in the shape of the thrombus. It turns out later that shape alone is not sufficient and the context/ location is also an important factor. Very strong shape detectors are neural networks, and using these lead to the first results for thrombus detection.

4.3.1 Multi-Directional U-Net

The U-Net (section 2.5.1) is a very successful method for bio-medical image segmentation. Thus it was also tested for thrombus segmentation. The models are trained from scratch on the CHSF stroke data-set. The first trained U-Nets didn't deliver even remotely useful results. It turned out that the class imbalance between thrombus and other tissues was too massive. This lead to the sampling scheme described in section 4.1.1 which mostly fixed the class imbalance. With this sampling the U-Net started to learn that dark thin objects have to be segmented, resulting in a model which segmented all dark areas like veins. And there are so many dark areas on susceptibility weighted angiography (SWAN) (see figure 4.7 a)) that this is no useful segmentation either. Even training the model with a *leave-one-out* cross validation scheme [107] didn't help. So it was clear that the

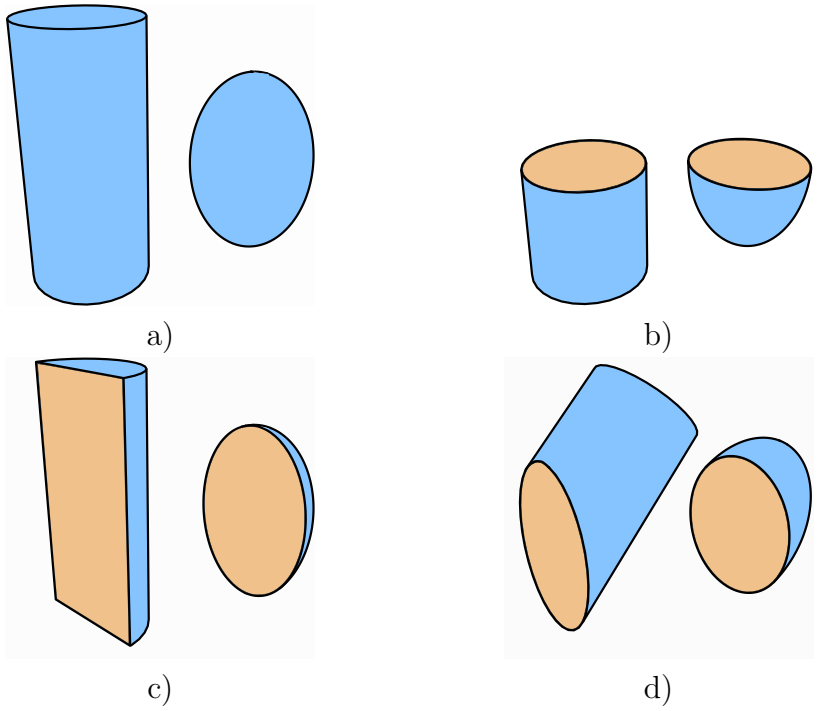
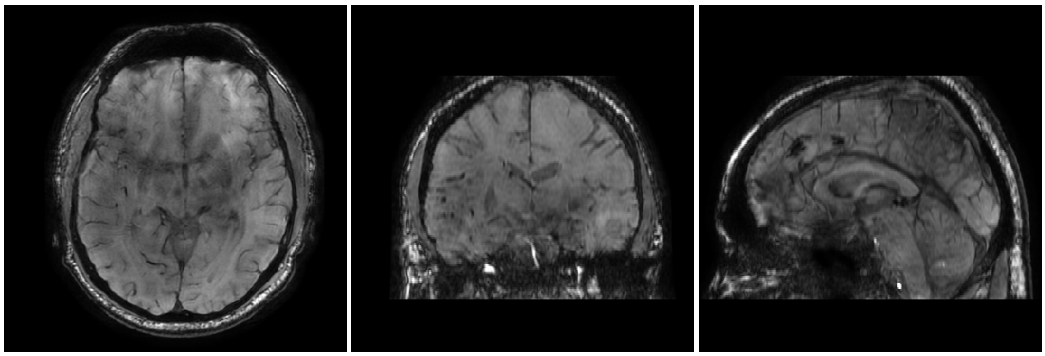


Figure 4.5 A tube, representing a vein, and an ellipsoid, representing a thrombus, are cut along different planes b)-c). Both veins and thrombi appear black on SWAN and a slice of the SWAN represents a cut through their respective geometries. Depending on the alignment of thrombus and arteries with respect to the slice they may look identical, see b) and d). Only on certain cuts their different shape becomes apparent c). This illustrates why a pure 2D approach can never reliably identify the thrombus and the 3D notion is required.

training data-set was insufficient for the U-Net model. This has been solved by creating a new training data-set with a special focus on the areas where the already trained model has problems.

To force the model to focus on the differences between thrombus and dark areas a new training set was sampled as follows: All patients in the training set were segmented with the previously trained U-Net model. From this segmentation false positives were noted. The new training set is sampled very similar to the previous sampling scheme, just that the patches with "likely false positive objects" are replaced by patches with known false positive objects. Then the already trained model is retrained on new training set. This improved the model to the point that it learned to segment dark round objects with very high sensitivity. But the problem is that on SWAN there are many dark round objects. The resulting U-Net segmented an average of



a) b) c)

Figure 4.6 SWAN slices of the three planes along which the three different U-Net models work: axial a), coronal b) and sagittal c).

133.6 false positive objects per patient. A sample prediction of this model is seen in figure 4.7 d).

Consultation with the neurologists of the CHSF revealed that the context available from a single slice of the SWAN is not enough to identify a thrombus. For example a vein with flow perpendicular to the slice produces a black spot indistinguishable from a thrombus (compare figure 4.5). This excuses the U-Net for its bad performance. To improve this segmentation results the three dimensional context needs to be included. For this three separate U-Nets are trained (with the retraining method above) on different orientations of the MRI. One is trained on axial slices, one on coronal slices and one on sagittal slices (see figure 4.6).

The three models provide three different scores s_1, s_2, s_3 for each voxel (figure 4.7 d)-f)) which need to be merged for a final decision. A simple majority vote would underestimate the size of the thrombus. Thus the label merging is split into two parts. At first possible thrombi are extracted and in the second part these candidates are classified into thrombus and non thrombus.

For the first part a new volume $\mathcal{V}$ is created where each voxel is assigned the maximum $\max(s_1, s_2, s_3)$ out of the three scores. $\mathcal{V}$ is then thresholded at a value of 0.4 and the resulting binary map is divided into connected components. To decide whether one component should be a thrombus or not, the scores (s_1, s_2, s_3) are used again. If all three scores are bigger than 0.7 for one voxel in the component, *i.e.* $s_1 > 0.7 \wedge s_2 > 0.7 \wedge s_3 > 0.7$, then the component is validated as a thrombus (figure 4.7 c)).

This architecture merging multiple U-Nets along multiple directions, the

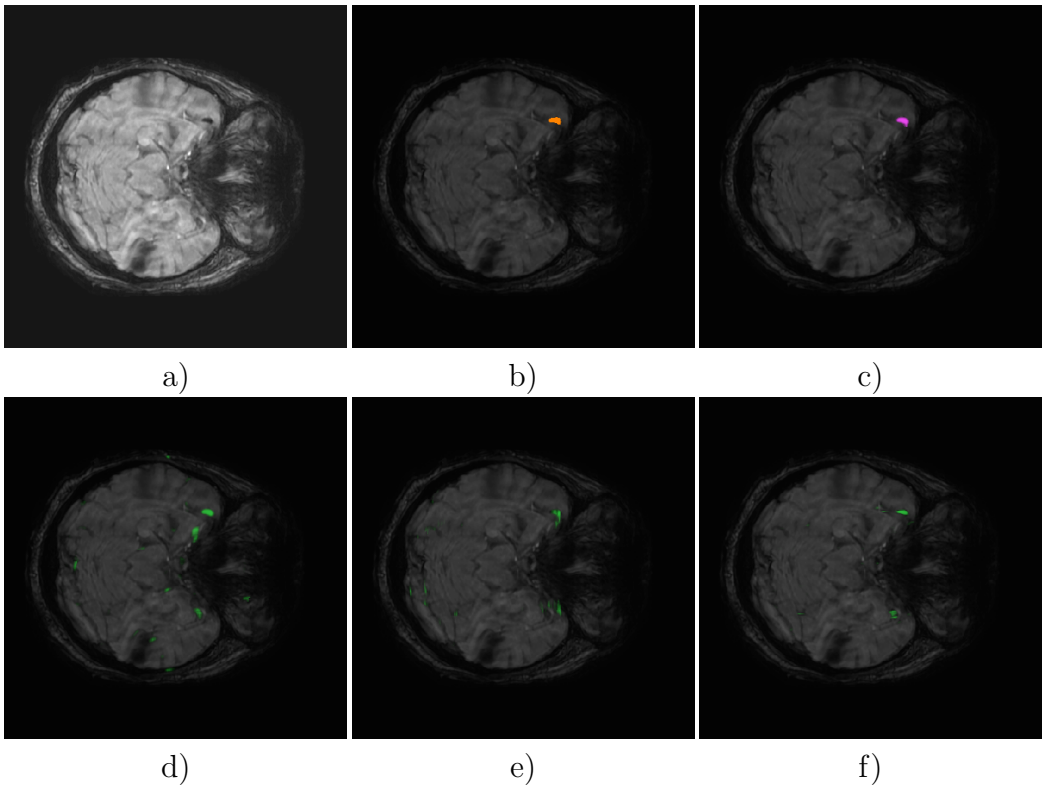


Figure 4.7 Results for the multi-directional U-Net. A SWAN image a) showing a thrombus. The thrombus location is given by the manual segmentation b). The segmentation from the merged predictions c) is very close to the ground truth. d), e) and f) are the predictions from the U-Nets in axial, coronal and sagittal direction. Each of the U-Net identifies a couple of the dark spots as a thrombus. But only the true thrombus is found by all three U-Nets, the other spots are always discarded by at least one U-Net. It is also visible that the size of the thrombus is underestimated by two of the U-Nets which made the first part of the label merging step necessary.

multi-directional U-Net, achieves a detection rate of 77.4% with a Dice coefficient of 0.415. On average it still detects 1.26 false positive objects. The three U-Nets can only capture information along three planes which is still not enough. Adding more models which work on even more directions should yield better results. Nevertheless this is a respectable first result and notably the first thrombus segmentation method from MRI added to literature [108]. But the thrombus segmentation will be improved by the following true 3D method which can leverage the whole context.

4.3.2 Mask R-CNN

A different CNN based architecture was tested by Markus Eppelsheimer in his master's thesis [109] on the CHSF stroke data-set. He used a mask R-CNN. This method comes from the semantic segmentation field and was designed to segment objects like cars and animals on photos. The architecture itself is rather complicated. It uses a Res-Net 101 [73] to extract features from the images. This is a pre-trained network which has been trained on a large data-base of real world images. The core idea of using a pre-trained network is that the features for something as complex as real world images are also present in medical images, *i.e.* lines, curves, dots, texture patters etc.. The features from the Res-Net are then input to another network (the so called region proposal network) which proposes regions of interest on the image where objects to segment may be present. The Res-Net features from the regions of interest are then scaled to a pre-defined resolution by the so called RoI align layer and then passed into a small CNN, the network head, for pixel wise segmentation.

Only the region proposal network, the RoI align layer and the network head are trained and have a relatively low amount of parameters, whereas the pre-trained Res-Net is not modified. This allows to profit of the feature extraction power of a large network as the Res-Net 101 while not having to train it.

The Mask R-CNN was trained on entire slices of the SWAN instead of the patches used for the other methods. Eppelsheimer evaluated the results of the Mask R-CNN slice by slice. To make them comparable to the results of this thesis they were reevaluated using the same method as for the other architectures tested in this thesis. The results are integrated with the other methods in tables 4.3 and 4.4. It turns out that its results are similar to the U-Net in that it finds all black spots on the image but is unable to differentiate between the thrombus and other black objects. This confirms the observation from the U-Net that a 2D method is unable to segment the thrombus reliably.

4.3.3 Logic LSTM

The logic LSTM described in section 3.2.3 has been designed for the task of thrombus segmentation. It has been tested on synthetic examples for its capacities to distinguish similar objects depending on the context in an very imbalanced data-set. It performs well on the task of thrombus segmentation,

too. To show the advantages of the logic LSTM it is compared to a 3D U-Net [102] which is trained under identical conditions.

The patients from the stroke data-set which show a thrombus are split into two halves, the first serves as training set and the second as test set. One patient from the training set is set aside as validation set. For training patches are sampled according to the sampling scheme in section 4.1.1. The learning rate is set to 0.001. Two logic LSTM models (Th 1 and Th 2) and the 3D U-Net are trained with the configurations given in table 4.2.

Table 4.2 Configurations of the two logic LSTM models and the 3D U-Net model trained for thrombus segmentation

Model	Patch Size	Batch Size	m	n_{4_c}	$n_{4_{\text{Input}}}$	Modalities
Th 1	$64 \times 64 \times 14$	10	4	3	16	SWAN
Th 2	$64 \times 64 \times 14$	10	4	3	16	ToF SWAN Phase
3D U-Net Th	$64 \times 64 \times 16$	4	-	-	32	SWAN

To obtain the final segmentation the probability maps produced by the networks were thresholded at a value of 0.5. Also the models Th 1 and Th 2 were merged by multiplying their probability maps and thresholding it at a value of 0.25. The segmentation results are given in table 4.3. It turns out that the biggest problem is to actually find the thrombus because there are many false positives. Thus the following discussion is focused on reducing the false positives and the actual segmentation quality (the Dice coefficient) is of minor interest.

The 3D U-Net architecture is almost identical to the U-Net used above, just that 3D convolutions are used on 3D inputs instead of 2D. It turns out that the 3D U-Net shares the problems of the U-Net. It only manages to detect black spots but it doesn't manage to differentiate between thrombus and non thrombus. On average the 3D U-Net detects 208 objects on the test set which is worse than the U-net. Certainly this could be improved by using the same cross validation and retraining scheme as for the U-Net. But one 3D U-Net model already needs around 4 days to train on an NVIDIA Tesla V100 GPU so this is not feasible.

In contrast the logic LSTM model Th 1 produces less than 30 false positives despite being trained under the same conditions. This already shows that the logic LSTM is a more robust model. Adding other MRI modalities doesn't really improve the results. The model Th 2 has a lower number of false

positives but has a lower detection rate. But what is interesting is that the models Th 1 and Th 2 detect different false positives. This observation lead to the idea to use the two models as ensemble (Th 1 & Th 2). The merged model only has 3.3 false positive objects on average but the detection rate drops to 0.93. This is already a largely better result than the multi-directional U-Net approach (section 4.3.1).

From these results it is clear that the thrombus detection is a very difficult problem and the SWAN alone is not sufficient. Additional clues for the location of the thrombus used by the medical doctors are their anatomical knowledge combined with the patients' symptoms. The patients' symptoms are not really retrievable in a retrospective study and currently cannot be retained past the anonymisation step. But there is another clue which is available: the location of the lesion which can be acquired from DWI (section 4.4). Unfortunately the lesion is often outside the patch size chosen for our models and increasing the patch size is not an option due to constraints in computational power. What is possible is to segment the lesion separately and use it in a post processing step to filter the false from the true positives.

It turns out that adding the lesion indeed improves the results significantly. The thrombus needs to be in the blood stream which supplies the lesion region. For all patients in the data base this is in front and below of the lesion, as well as between the outer border of the lesion and the mid-sagittal plane. To avoid calculating the position of the mid-sagittal plane it is assumed that the thrombus is not further inward than 50 voxels from the inner edge of the lesion. This excludes the unaffected hemisphere and is far enough for all thrombi. Sometimes the thrombus can be inside the lesion, so the selection rule with respect to the lesion is:

Thrombus selection by lesion position The valid location for the thrombus is inside a cube given by the lesion location. It is below the upper edge of the lesion, in front of the rear edge of the lesion and between the outer edge of the lesion and 50 voxels inward of the inner edge of the lesion. The inner edge is the one closest to the mid sagittal plane (which can be replaced by the centre of the MRI volume for calculation purposes).

This selection allows to discard many false positives. Using the lesion the number of false positives drops massively for all models (table 4.3 lower part). But the relative performance of the models stays the same. The U-Net and 3D U-Net still detect more than 40 false positives which is far from a reliable detection. The merged logic LSTM model Th 1 & Th 2 improves to only 1.5 false positive objects with a detection rate of 93%. This would be good enough to use it in a semi automatic fashion where the medical doctor

would only need to decide between two possible thrombi. This shows the importance of the lesion as an indicator for the thrombus location. Also the lesion is such a strong feature that the threshold for the probability map becomes non-relevant. For the merged logic LSTM model Th 1 & Th 2 the threshold could be lowered to 0.02 without really affecting the number of false positives. The lowered threshold provided a slightly better Dice-coefficient.

The average size of the false positive components is certainly below 100 voxels for all models. However the smallest thrombus in the data-set has a volume of 350 voxel. Of course this observation can be used to improve the segmentation results. Discarding all objects which are smaller than 300 voxels yields the results in table 4.4. Most false positives vanish. The results from the U-Net models are still mediocre while the logic LSTM delivers almost perfect results. The merged model Th 1 & Th 2 even manages to have zero false positives. This can be called a true automatic thrombus detection! Together with the lesion as additional feature the threshold can be lowered which leads to a detection rate of 100%. This is a perfect result.

To emphasise the difficulty of the thrombus segmentation task the segmentation on two particularly difficult patients is shown in figure 4.8. The first row shows a patient which has been trembling during the MRI acquisition and thus the image is very blurry. To a human eye the thrombus is barely recognisable. Nevertheless the logic LSTM identifies the thrombus. The second row shows a patient with many micro bleeds. Micro bleeds are little haemorrhages of the same size as the thrombus which don't provoke any symptoms. As they are constituted of clotted blood they are identical to a thrombus in terms of appearance on MRI. The only way to identify a micro bleed is by the context, *i.e.* its location with respect to the arterial tree and the lesion. Still the logic LSTM can tell the difference where other methods fail.

These experiments show that the thrombus segmentation from SWAN is a very difficult task. Established segmentation methods (U-Net and 3D U-Net) are very good at pattern recognition. But it turns out that the pattern alone is not sufficient as there are many objects with identical patterns. A more advanced model, the logic LSTM, shows that it is capable to incorporate context information which largely improves its results over the U-Net, 3D U-Net as well as the multi-directional U-Net. Even the improved models still produce some false positives. Post-processing of the segmentation with the knowledge of the lesion location and minimum size of the thrombus allows to achieve a perfect thrombus detection. The thrombus is detected in 100% of all patients with no false positives. This is the first reliable thrombus segmentation. Thus the logic LSTM can be truly used for an automatic

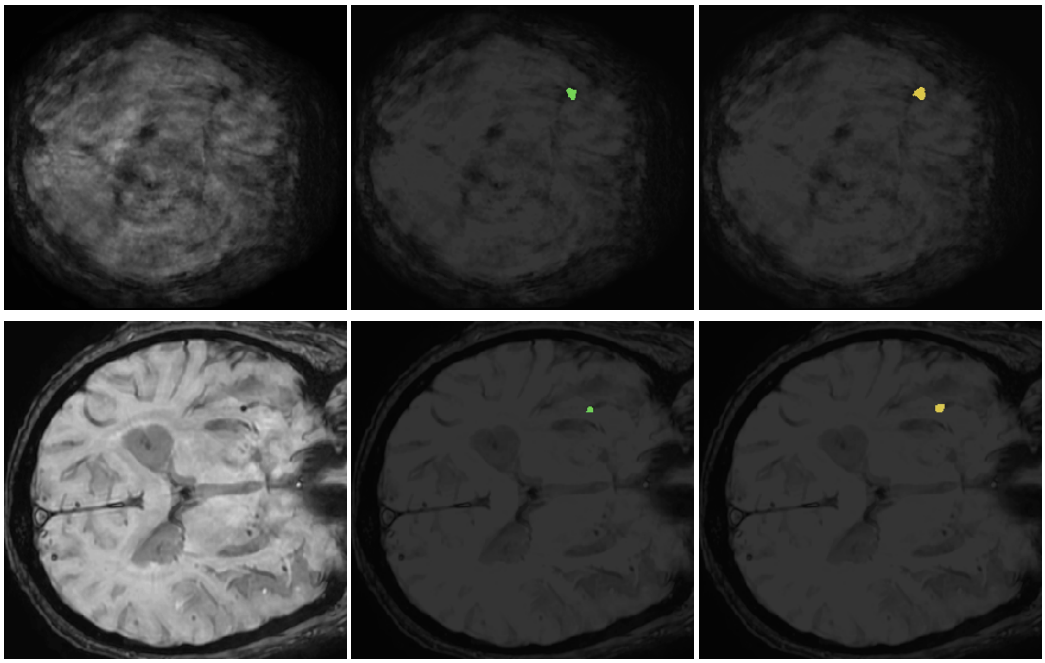


Figure 4.8 Two examples of thrombus segmentation by the logic LSTM. The first column is the SWAN, the second column the ground truth and the last column the segmentation given by the ensemble of models Th 1 & Th 2. The first row shows a patient with strong movement artifacts, but despite the excessive blurring of the image the logic LSTM still finds the thrombus. The second patient shows that the thrombus is very similar to many structures on SWAN.

detection of the thrombus. Of course this approach needs to be verified with a larger data-set but is very promising and certainly a candidate for clinical application.

The logic LSTM also has a number of other advantages. It uses less than 3% of the parameters of established segmentation methods, it can be trained with an even smaller number of samples, it is a more robust model and it is not only an object detector but can also take into account simple logical conditions. Notably it is very good at telling apart very similar objects. This makes it an interesting method for other applications as well.

Table 4.3 Segmentation results for the thrombus. The lower part gives the results obtained by using the lesion location as additional feature. The columns are the Dice coefficient, the mean number of false positive objects "FP" and their average size "FP Size", the mean number of false negative objects "FN" and their average size "FN Size" and the detection rate D_r.

Model	Dice	FP	FP Size	FN	FN Size	D_r
Thrombus detection						
U-Net Th	0.08	133.6	94.0	0.14	353.8	0.96
Mask R-CNN	0.056	229.7	215	1.48	643.5	1.0
3D U-Net Th	0.035	208.3	83.8	0.25	263.9	0.89
Th 1	0.16	29.8	76.3	0.25	121.1	1.0
Th 2	0.10	22.2	38.1	0.43	240.6	0.89
Th 1 & Th 2	0.19	3.3	29.6	0.39	239.5	0.93
Thrombus detection using lesion						
U-Net Th	0.18	47.7	93.5	0.14	353.8	0.96
Mask R-CNN	0.17	67.8	194.4	1.48	643.5	1.0
3D U-Net Th	0.14	44.0	81.9	0.25	263.9	0.89
Th 1	0.28	10.7	62.6	0.25	3121.1	1.0
Th 2	0.15	6.6	39.7	0.46	242.6	0.89
Th 1 & Th 2	0.22	1.5	24.8	0.43	241.9	0.93

4.4 Automatic Lesion Segmentation

While the thrombus is only a small dark spot on SWAN, the lesion is the damaged area and can be almost as big as a hemisphere. Just the larger size makes it easier to segment and lesion segmentation has already been studied as explained in section 1.3.5. Please recall that all those studies have been in the acute phase or even later. The stroke data-set used here has patients in the hyper acute phase of stroke. Which means that the lesion is developing, can be small and has no well defined border (figure 1.9). This makes it a more difficult task than the cases which have been studied already in literature. DWI has numerous artefacts which are as bright as the lesion, which is not discussed in most studies. In the hyper-acute phase of stroke some artefacts are visually indistinguishable (through the similar size and equally fuzzy borders) from the lesion and thus artefact detection becomes a major concern. Thanks to the innovative lesion enhancement pre-processing (section 4.2.4) most artefacts can already be eliminated but some still remain.

Table 4.4 Segmentation results for the thrombus where objects of less than 300 voxels volume have been deleted. The lower part gives the results obtained by using the lesion location as additional feature. The columns are the Dice coefficient, the mean number of false positive objects "FP" and their average size "FP Size", the mean number of false negative objects "FN" and their average size "FN Size" and the detection rate D_r.

Model	Dice	FP	FP Size	FN	FN Size	D_r
Thrombus detection, size > 300						
U-Net Th	0.08	8.4	545.6	0.14	353.8	0.96
Mask R-CNN	0.056	40.1	810.2	1.48	643.5	1.0
3D U-Net Th	0.035	14.8	516.8	0.25	263.9	0.89
Th 1	0.16	1.4	459.3	0.25	121.1	1.0
Th 2	0.10	0.14	369.25	0.43	240.6	0.89
Th 1 & Th 2	0.19	0.0	-	0.39	239.5	0.93
Thrombus detection using lesion and size > 300						
U-Net Th	0.19	3.1	515.4	0.14	353.8	0.96
Mask R-CNN	0.17	11.0	723.0	1.48	643.5	1.0
3D U-Net Th	0.14	2.5	524.6	0.25	263.9	0.89
Th 1	0.28	0.36	386.6	0.25	121.1	1.0
Th 2	0.15	0	-	0.46	242.6	0.89
Th 1 & Th 2	0.28	0.07	389.5	0.32	196.9	1.0

4.4.1 Fuzzy Clustering

Fuzzy clustering is a popular unsupervised method for medical image segmentation. In the context of this work the unsupervised methods have been studied by Andrea Roncoli. He compared the methods by Tasi et al. [69] and Muda et l. [68] as well as basic fuzzy C-means (FCM) and improved upon these methods. Essentially the method by [68] is completely useless for emerging lesions as it only uses FCM to find a threshold and thus only works with well defined bright lesions while also segmenting all possible artefacts. Tsai et al. [69] provide a method for detecting artefacts and while the numerical values for their parameters had to be adjusted to make it work on the CHSF stroke data-set it detects some artefacts. Still the results are mediocre.

Andrea Roncoli could improve the method. His FCM based method is described in his master's thesis [110]. The Dice coefficients achieved with these

unsupervised methods are given in table 4.5. For an unsupervised method Roncoli's result is not bad and actually comparable to the latest published result using a CNN [65] with a Dice score of 0.61.

Table 4.5 Dice score for the lesion segmentation task for the three fuzzy clustering methods on the CHSF stroke data-set.

Mehtod	Muda et al.	Tsai et al.	Roncoli et al.
Dice	0.46	0.53	0.63

4.4.2 U-Net

Clustering methods have their advantages in being very small and fast models. But overall better results are expected with supervised deep learning methods. As for the thrombus the first tried method is the U-Net. It was trained on 64×64 patches using DWI and the enhanced lesion image (section 4.2.4) with a batch size of 100, 32 start features and with a *leave-one-out* cross validation scheme [107]. With a Dice score of 0.59 it achieved the performance which is expected by a CNN on this kind of problem ([65] achieved a Dice of 0.61). This is still below the inter-observer agreement of 0.68 and notably the U-Net has trouble with artefacts.

4.4.3 Logic LSTM

The logic LSTM has shown on the thrombus segmentation task that it is able to tell apart very similar objects. As the lesion segmentation is an easier task than the thrombus segmentation the logic LSTM really excels at this task. The data-set is divided in half, like for the thrombus segmentation, into a training and test set with one patient set aside as validation set. The learning rate is set to 0.001. Two logic LSTM models (Le 1 and Le 2), an U-Net and a 3D U-Net are trained to segment the lesion with the configurations given in table 4.6.

The segmentation results of all four networks are given in table 4.7. The 3D U-Net once again suffers from the small training set while the U-Net generally learns to detect the lesion but fails in some cases. The logic LSTM model Le 2 detects the lesion in every patient. A few false positive objects of minor size are found. Upon manual inspection it turns out that these are very close to the main lesion and might even be part of the lesion, depending

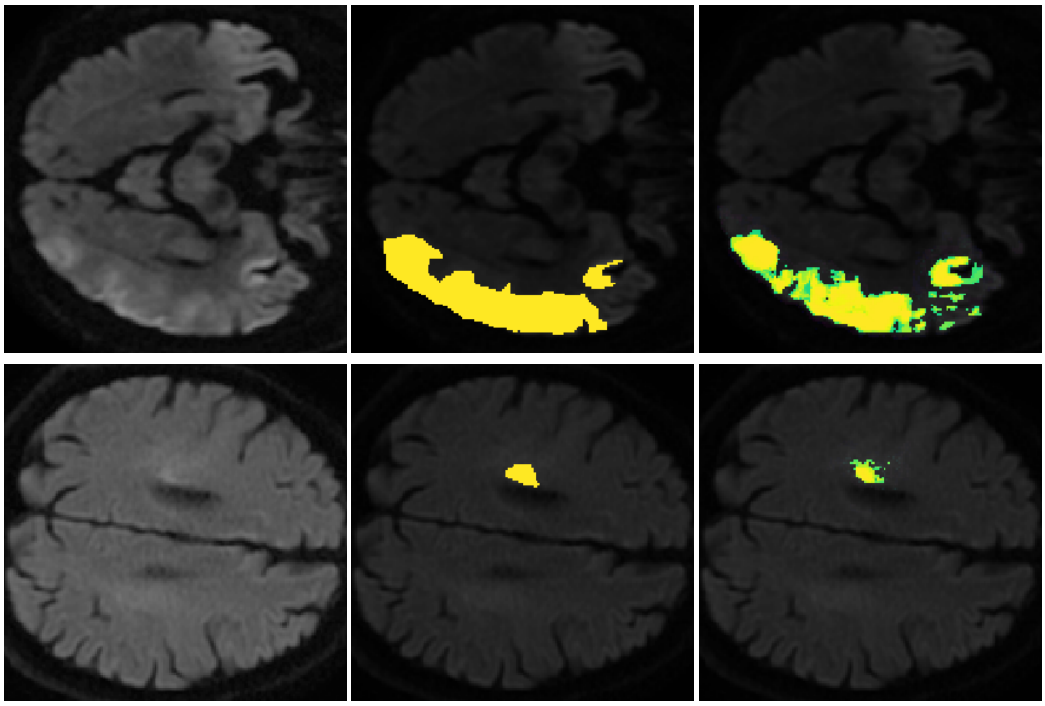


Figure 4.9 Two examples of developing lesions in the hyper-acute phase of stroke (first column). The lesion is currently growing and has no clearly defined borders but rather shows gradients towards the normal tissue. The second column shows the ground truth and the third column the lesion probability given by the model Le 2. Please note that our algorithm correctly identified the contra-lateral artifact visible in the first row.

on definition. So it is an almost perfect result. However by merging the two models Le 1 and Le 2 the average number of false positives can be reduced to 0.6. As the false positives constituted an insignificant amount of voxels anyway the Dice coefficient only changes at the fourth digit after the decimal point. The logic LSTM achieves a Dice coefficient of 0.69 which corresponds to the inter-observer agreement of 0.68. So this result is a segmentation of the lesion on human level. Notably it is the first lesion segmentation which achieves this performance and a better result cannot be achieved on this type of data-base. Two especially difficult cases are displayed in figure 4.9. They show the gradual borders of the developing lesion and contra lateral artefacts which only trained experts can identify as such.

Table 4.6 Configuration of the networks for the lesions segmentation task. E is the enhanced image from section 4.2.4.

Model	Patch Size	Batch Size	m	n_{4_c}	$n_{4_{\text{Input}}}$	Modalities
Le 1	$128 \times 128 \times 16$	4	4	4	32	DWI E
Le 2	$256 \times 256 \times 14$	2	4	4	32	DWI E
U-Net	$128 \times 128 \times 1$	100	-	-	16	DWI E
3D U-Net	$128 \times 128 \times 16$	4	-	-	32	DWI E

Table 4.7 Segmentation results for the lesion. The columns are the Dice coefficient, the mean number of false positive objects "FP" and their average size "FP Size", the mean number of false negative objects "FN" and their average size "FN Size" and the detection rate D_r.

Model	Dice	FP	FP Size	FN	FN Size	D_r
U-Net	0.59	6.8	70.4	1.3	61.9	0.96
3D U-Net	0.34	31.4	231.4	1.7	19.4	1.0
Le 1	0.64	2.2	4.3	2.2	62.4	1.0
Le 2	0.69	3.4	35.4	1.4	20.0	1.0
Le 1 & Le 2	0.69	0.6	44.7	1.8	51.3	1.0

Chapter 5

Conclusion and Perspectives

5.1 Summary

The starting point for this thesis was the detection of thrombus and lesion from stroke MRI using machine learning methods. For this goal a data-set with images of 65 patients was provided by the CHSF neurology group. After exploring the details of the acquisition technique, MRI, and studying the medical implications of a stroke it was possible to decide which of the images could be used for thrombus detection and which for lesion detection.

The thrombus segmentation task proved to be particularly difficult. Traditional machine learning and image processing methods were without success. After studying the problem it was clear that texture or statistical features are not sufficient and the usage of a shape detector was necessary. Convolutional neural networks are currently the best shape detectors and the U-Net, which is specialised on bio-medical images, was used. Training the U-Net on the Stroke data-set was difficult and resulted in the development of a new sampling method to relieve the class imbalance. This lead to the insight that 2D methods can find the thrombus but will inevitably produce hundreds of false positives. Of course trying a 3D U-Net would have been an idea but the NVIDIA 1070 GPU available at that point did not have enough RAM to actually train one. Later, when an NVIDIA Tesla V100 GPU was available it turned out that the 3D U-Net is not powerful enough for the thrombus segmentation task. Thus the U-Net was extended to a multi-directional U-net by merging three separate U-Nets which work on the axial, coronal and sagittal slices respectively. Although not true 3D this allowed to find the thrombus in 77.4% of all patients but on the downside with a small number of false positives. Inspired by the way medical doctors find the thrombus by looking at the series of slices a recurrent network architecture, the logic

LSTM, was designed. It processes the MRI volume slice by slice and has a memory through which it can transfer features from slice to slice. This allows to use the full 3D context while using only a 2D method. The logic LSTM alone achieves results far superior to the U-Net, and when used in an ensemble of two it can reliably find the thrombus with a detection rate of 100% and produces no false positives. Furthermore the logic LSTM is more efficient than the U-Net, using only 3% of the U-Nets parameters. So this work presents the first reliable thrombus detection from stroke MRI something which could not be achieved by competing architectures.

With the logic LSTM giving such good results for the thrombus it is no surprise that it also excelled at the easier task of lesion segmentation. A new pre-processing step makes the lesion much more visible and established segmentation methods have no problem finding it. However the lesion in the hyper-acute phase of stroke is still difficult to delineate precisely because it is gradually developing and has no well defined borders. The logic LSTM improves upon this point and contributes the first lesion segmentation with human level performance. No other method before achieved a Dice score as high as the inter-observer agreement on the stroke lesion segmentation task.

Not only the stroke MRI segmentation tasks have been solved, the development of the logic LSTM also brought some theoretical insights in the field of neural network design. With the transfer block the standard method for increasing receptive fields, multi-resolution pyramids, has a new competitor which uses much less parameters. Also the convolutional LSTM has undergone a detailed analysis and its weaknesses have been identified and leveraged. Resolving these lead to the logic LSTM.

5.2 Future Work

The new solutions to the stroke segmentation problem are very promising so the future plan is to validate these on a larger data-set and to integrate it into a software for clinical application. Hopefully this will become a valuable tool for the treatment decision for stroke patients. But it does not need to stay a tool for measuring the lesion and thrombus size. As it has been shown that the thrombus and lesion size are good predictors for the patient outcome after treatment this is a candidate for automation. Possibly the patient outcome can be predicted with respect to the treatment option. This could be helpful to choose the most favourable treatment for the patient.

On the theoretical side there are many interesting aspects to investigate. The methods for increasing the receptive field for a neuron deserve more attention. The proposed transfer block increases the receptive field in only one step and it was shown that it performs comparable to a fully fledged multi resolution method. But the features extracted by the transfer block are only local, combined with a spatial information given by the different window sizes. It would be interesting to see how the local plus spatial information compares to multi-resolution features in different use cases and if they are truly interchangeable. Also there may be other ideas on how to increase the receptive field. From an applied point of view the transfer block offers small, *i.e.* cheap, networks which could unblock applications of deep learning in areas where computational power is limited like automobile applications or real-time clinical use cases. Also the "logic" part of the logic LSTM provides opportunities for studying the information processing in recurrent networks. The logic LSTM itself is a first step in connecting visual information with logic processing which surpasses the element of correlation and statistics which is the basis of current CNNs. Maybe the logic LSTM can master tasks which go beyond segmentation such as controlling a (industrial) process or be the basis of such an architecture.

Bibliography

[1] Thijs, V. N., Lansberg, M. G., Beaulieu, C. et al. 'Is early ischemic lesion volume on diffusion-weighted imaging an independent predictor of stroke outcome?' *Stroke*, volume 31(11):2597–2602, November 2000. doi:10.1161/01.str.31.11.2597

[2] Legrand, L., Naggara, O., Turc, G. et al. 'Clot burden score on admission T2*-MRI predicts recanalization in acute stroke'. *Stroke*, volume 44(7):1878–1884, July 2013. doi:10.1161/strokeaha.113.001026

[3] Kamalian, S., Morais, L. T., Pomerantz, S. R. et al. 'Clot length distribution and predictors in anterior circulation stroke'. *Stroke*, volume 44(12):3553–3556, 2013. ISSN 0039-2499. doi:10.1161/STROKEAHA.113.003079
URL `http://stroke.ahajournals.org/content/44/12/3553`

[4] Liebeskind, D. S. 'Collateral circulation'. *Stroke*, volume 34(9):2279–2284, 2003. ISSN 0039-2499. doi:10.1161/01.STR.0000086465.41263.06
URL `http://stroke.ahajournals.org/content/34/9/2279`

[5] Lee, J. M., Grabb, M. C., Zipfel, G. J. et al. 'Brain tissue responses to ischemia.' *The Journal of clinical investigation*, volume 106 6:723–31, 2000

[6] Baron Jean-Claude and Moseley Michael E. 'For how long is brain tissue salvageable? imaging-based evidence'. *Journal of Stroke and Cerebrovascular Diseases*, volume 9(6):15–20, February 2018. ISSN 1052-3057. doi:10.1053/jscd.2000.18910;10.1053/jscd.2000.18910

[7] Kidwell, C. S., Alger, J. R. and Saver, J. L. 'Beyond mismatch'. *Stroke*, volume 34(11):2729–2735, 2003. ISSN 0039-2499. doi:10.1161/01.STR.0000097608.38779.CC
URL `http://stroke.ahajournals.org/content/34/11/2729`

[8] Dohmen C., Galldiks N., Bosche B. et al. 'The severity of ischemia determines and predicts malignant brain edema in patients with large middle cerebral artery infarction'. *Cerebrovascular Diseases*, volume 33(1):1–7, 2012. ISSN 1015-9770. doi:10.1159/000330648

[9] Hwang Jee-Yeon, Gertner Michael, Pontarelli Fabrizio et al. 'Global ischemia induces lysosomal-mediated degradation of mTOR and activation of autophagy in hippocampal neurons destined to die'. *Cell Death And Differentiation*, volume 24:317, December 2016. doi:10.1038/cdd.2016.140;10.1038/cdd.2016.140
URL `https://www.nature.com/articles/cdd2016140#supplementary-information`

[10] Jeon Jaepyo, Sun Guanghua, Tian Jinbin et al. 'The role of TRPC channels in ischemic neuronal cell death'. *Biophysical Journal*, volume 112(3):408a, November 2017. ISSN 0006-3495. doi:10.1016/j.bpj.2016.11.2207;;10.1016/j.bpj.2016.11.2207

[11] Balaganapathy, P., Baik, S.-H., Mallilankaraman, K. et al. 'Interplay between notch and p53 promotes neuronal cell death in ischemic stroke'. *Journal of Cerebral Blood Flow & Metabolism*, volume 38(0):1781–1795, June 2017. doi:10.1177/0271678x17715956
URL `https://doi.org/10.1177/0271678X17715956`

[12] Heiland, S. Disturbed Brain Perfusion, pp. 103–116. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-27738-5. doi:10.1007/3-540-27738-2

[13] Lin, T.-C., Lee, J.-D., Lin, Y.-H. et al. 'Timing of symptomatic infarct swelling following intravenous thrombolysis in acute middle cerebral artery infarction: A case–control study'. *Clinical and Applied Thrombosis/Hemostasis*, volume 23(7):814–820, 2017. doi:10.1177/1076029616659693
URL `https://doi.org/10.1177/1076029616659693`

[14] Taylor, B., Lopresti, M., Appelboom, G. et al. 'Hemicraniectomy for malignant middle cerebral artery territory infarction: an updated review'. *Journal of Neurosurgical Sciences*, (59(1)):73–8, March 2015

[15] Hacke Werner, Kaste Markku, Bluhmki Erich et al. 'Thrombolysis with alteplase 3 to 4.5 hours after acute ischemic stroke'. *New England Journal of Medicine*, volume 359(13):1317–1329, 2008. ISSN 0028-4793. doi:10.1056/NEJMoa0804656;10.1056/NEJMoa0804656

[16] Olindo, S., Chausson, N., Joux, J. et al. 'Fluid-attenuated inversion recovery vascular hyperintensity: An early predictor of clinical outcome in proximal middle cerebral artery occlusion'. *Archives of Neurology*, volume 69(11):1462–1468, November 2012. ISSN 0003-9942. doi:10.1001/archneurol.2012.1310

[17] Santos Emilie M.M., Dankbaar Jan Willem, Treurniet Kilian M. et al. 'Permeable thrombi are associated with higher intravenous recombinant tissue-type plasminogen activator treatment success in patients with acute ischemic stroke'. *Stroke*, volume 47(8):2058, August 2016
URL `http://stroke.ahajournals.org/content/47/8/2058.abstract`

[18] Yaghi, S., Eisenberger, A. and JZ, W. 'Symptomatic intracerebral hemorrhage in acute ischemic stroke after thrombolysis with intravenous recombinant tissue plasminogen activator: A review of natural history and treatment'. *JAMA Neurology*, volume 71(9):1181–1185, 2014. doi:10.1001/jamaneurol.2014.1210
URL `http://dx.doi.org/10.1001/jamaneurol.2014.1210`

[19] Smadja Didier, Chausson Nicolas, Joux Julien et al. 'A new therapeutic strategy for acute ischemic stroke'. *Stroke*, volume 42(6):1644, June 2011
URL `http://stroke.ahajournals.org/content/42/6/1644.abstract`

[20] Lapergue, B., Blanc, R., Gory, B. et al. 'Effect of endovascular contact aspiration vs stent retriever on revascularization in patients with acute ischemic stroke and large vessel occlusion: The aster randomized clinical trial'. *JAMA*, volume 318(5):443–452, August 2017. ISSN 0098-7484. doi:10.1001/jama.2017.9644

[21] Goyal Mayank, Menon Bijoy K, van Zwam Wim, H. et al. 'Endovascular thrombectomy after large-vessel ischaemic stroke: a meta-analysis of individual patient data from five randomised trials'. *The Lancet*, volume 387(10029):1723–1731, February 2018. ISSN 0140-6736. doi:10.1016/S0140-6736(16)00163-X;;10.1016/S0140-6736(16)00163-X

[22] Pfaff J., Herweh C., Pham M. et al. 'Mechanical thrombectomy of distal occlusions in the anterior cerebral artery: Recanalization rates, periprocedural complications, and clinical outcome'. *American Journal*

of Neuroradiology, volume 37(4):673, April 2016
URL `http://www.ajnr.org/content/37/4/673.abstract`

[23] Powers William J., Rabinstein Alejandro A., Ackerson Teri 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*, volume 49(3), January 2018
URL `http://stroke.ahajournals.org/content/early/2018/01/23/STR.0000000000000158.abstract`

[24] Yaghi, S., Willey, J. Z., Cucchiara, B. et al. 'Treatment and outcome of hemorrhagic transformation after intravenous alteplase in acute ischemic stroke: A scientific statement for healthcare professionals from the american heart association/american stroke association'. *Stroke*, volume 48(12), December 2017. doi:10.1161/str.0000000000000152

[25] Mazya, M., Egido, J. A., Ford, G. A. et al. 'Predicting the risk of symptomatic intracerebral hemorrhage in ischemic stroke treated with intravenous alteplase'. *Stroke*, volume 43(6):1524–1531, June 2012. doi:10.1161/strokeaha.111.644815

[26] Senior, K. 'Microbleeds may predict cerebral bleeding after stroke'. *The Lancet*, volume 359(9308):769, March 2002. doi:10.1016/s0140-6736(02)07911-4

[27] Wang, S., Lv, Y., Zheng, X. et al. 'The impact of cerebral microbleeds on intracerebral hemorrhage and poor functional outcome of acute ischemic stroke patients treated with intravenous thrombolysis: a systematic review and meta-analysis'. *Journal of Neurology*, volume 264(7):1309–1319, November 2016. doi:10.1007/s00415-016-8339-1

[28] Brott, T., Adams, H. P., Olinger, C. P. et al. 'Measurements of acute cerebral infarction: a clinical examination scale.' *Stroke*, volume 20(7):864–870, 1989. ISSN 0039-2499. doi:10.1161/01.STR.20.7.864
URL `http://stroke.ahajournals.org/content/20/7/864`

[29] P Adams, H., H Davis, P., Leira, E. et al. 'Baseline NIH stroke scale score strongly predicts outcome after stroke: a report of the trial of ORG 10172 in acute stroke treatment (TOAST)'. volume 53:126–31, July 1999

[30] von Kummer, R. and Dzialowski, I. 'Imaging of cerebral ischemic edema and neuronal death'. *Neuroradiology*, volume 59(6):545–553, May 2017. doi:10.1007/s00234-017-1847-6

[31] Heo Ji Hoe, Kim Kyeonsub, Yoo Joonsang et al. 'Computed tomography-based thrombus imaging for the prediction of recanalization after reperfusion therapy in stroke'. *Journal of Stroke*, volume 19(1):40–49, January 2017. ISSN 2287-6391 2287-6405. doi:10.5853/jos.2016.01522
URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5307933/

[32] Levitt, M. H. 'Spin dynamics: Basics of nuclear magnetic resonance', 2008

[33] Lauterbur P. C. 'Image formation by induced local interactions: Examples employing nuclear magnetic resonance'. *Nature*, volume 242:190, March 1973. doi:10.1038/242190a0;10.1038/242190a0

[34] Mansfield, P. 'Multi-planar image formation using NMR spin echoes'. *Journal of Physics C: Solid State Physics*, volume 10(3):L55, 1977
URL http://stacks.iop.org/0022-3719/10/i=3/a=004

[35] Le Bihan, D. 'Diffusion MRI: what water tells us about the brain'. *EMBO Molecular Medicine*, volume 6(5):569–573, 2014. ISSN 1757-4676. doi:10.1002/emmm.201404055
URL http://embomolmed.embopress.org/content/6/5/569

[36] Kiselev, V. 'Fundamentals of diffusion MRI physics'. volume 30, March 2017

[37] Le Bihan, D., Poupon, C., Amadon, A. et al. 'Artifacts and pitfalls in diffusion MRI'. *Journal of Magnetic Resonance Imaging*, volume 24(3):478–488, 2006. ISSN 1522-2586. doi:10.1002/jmri.20683
URL http://dx.doi.org/10.1002/jmri.20683

[38] Jones, D. K. and Cercignani, M. 'Twenty-five pitfalls in the analysis of diffusion MRI data'. *NMR in Biomedicine*, volume 23(7):803–820, 2010. ISSN 1099-1492. doi:10.1002/nbm.1543
URL http://dx.doi.org/10.1002/nbm.1543

[39] Hajnal, J., De Coene, B., D. Lewis, P. et al. 'High signal regions in normal white matter shown by heavily T2 weighted CSF nulled IR sequences'. volume 16:506–513, July 1992

[40] Smith Eric E. 'Leukoaraiosis and stroke'. *Stroke*, volume 41(10 suppl 1):S139, October 2010

[41] Kamran, S., Bates, V., Bakshi, R. et al. 'Significance of hyperintense vessels on FLAIR MRI in acute stroke'. *Neurology*, volume 55(2):265–269, 2000. ISSN 0028-3878. doi:10.1212/WNL.55.2.265
URL `http://n.neurology.org/content/55/2/265`

[42] Dumoulin, C. L., Cline, H. E., Souza, S. P. et al. 'Three-dimensional time-of-flight magnetic resonance angiography using spin saturation'. *Magnetic Resonance in Medicine*, volume 11(1):35–46, 1989. ISSN 1522-2594. doi:10.1002/mrm.1910110104
URL `http://dx.doi.org/10.1002/mrm.1910110104`

[43] Parker, D. L., Yuan, C. and Blatter, D. D. 'MR angiography by multiple thin slab 3D acquisition'. *Magnetic Resonance in Medicine*, volume 17(2):434–451, 1991. ISSN 1522-2594. doi:10.1002/mrm.1910170215
URL `http://dx.doi.org/10.1002/mrm.1910170215`

[44] Linn, J. 'High-resolution 3D imaging of the cerebral vasculature'. *SignaPULSE*, 2009

[45] Annamraju, R. B. V. R. and Vu, A. T. 'T2* weighted angiography (SWAN): T2* weighted non-contrast imaging with multi-echo acquisition and reconstruction'. In ESMRMB, 2008

[46] Roy, S., Butman, J. A. and Pham, D. L. 'Robust skull stripping using multiple MR image contrasts insensitive to pathology'. *NeuroImage*, volume 146:132–147, 2017

[47] Manjón, J. V. and Coupé, P. 'volBrain: An online MRI brain volumetry system'. *Frontiers in Neuroinformatics*, volume 10, July 2016. doi:10.3389/fninf.2016.00030

[48] Benou, A., Veksler, R., Friedman, A. et al. 'Ensemble of expert deep neural networks for spatio-temporal denoising of contrast-enhanced MRI sequences'. *Medical Image Analysis*, volume 42:145–159, December 2017. doi:10.1016/j.media.2017.07.006

[49] Klepaczko, A., Szczypiński, P., Deistung, A. et al. 'Simulation of MR angiography imaging for validation of cerebral arteries segmentation algorithms'. *Computer Methods and Programs in Biomedicine*, volume 137:293–309, December 2016. doi:10.1016/j.cmpb.2016.09.020

[50] Sampat, M. P., Wang, Z., Markey, M. K. et al. 'Measuring intra- and inter-observer agreement in identifying and localizing structures in medical images'. In 2006 International Conference on Image Processing. IEEE, October 2006. doi:10.1109/icip.2006.312367

[51] Liu, L., Ding, J., Leng, X. et al. 'Guidelines for evaluation and management of cerebral collateral circulation in ischaemic stroke 2017'. *Stroke and Vascular Neurology*, volume 3(3):117–130, May 2018. doi:10.1136/svn-2017-000135

[52] Egger, J., O'Donnell, T., Hopfgartner, C. et al. Graph-based Tracking Method for Aortic Thrombus Segmentation, pp. 584–587. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-89208-3. doi:10.1007/978-3-540-89208-3

[53] Olabarriaga, S. D., Rouet, J. M., Fradkin, M. et al. 'Segmentation of thrombus in abdominal aortic aneurysms from CTA with nonparametric statistical grey level appearance modeling'. *IEEE Transactions on Medical Imaging*, volume 24(4):477–485, April 2005. ISSN 0278-0062. doi:10.1109/TMI.2004.843260

[54] Qazi, E., Wilson, A. T., McDougall, C. et al. 'Abstract TP45: One threshold does not fit all: Hounsfield unit thresholds to segment clot on NCCT are patient specific'. *Stroke*, volume 47(Suppl 1):ATP45–ATP45, 2016. ISSN 0039-2499
URL `http://stroke.ahajournals.org/content/47/Suppl_1/ATP45`

[55] Santos, E. M. M., Marquering, H. A., Berkhemer, O. A. et al. 'Development and validation of intracranial thrombus segmentation on CT angiography in patients with acute ischemic stroke'. *PLOS ONE*, volume 9(7):1–8, July 2014. doi:10.1371/journal.pone.0101985
URL `https://doi.org/10.1371/journal.pone.0101985`

[56] Riedel, C. H., Jensen, U., Rohr, A. et al. 'Assessment of thrombus in acute middle cerebral artery occlusion using thin-slice nonenhanced computed tomography reconstructions'. *Stroke*, volume 41(8):1659–1664, 2010. ISSN 0039-2499. doi:10.1161/STROKEAHA.110.580662
URL `http://stroke.ahajournals.org/content/41/8/1659`

[57] Lesage, D., Angelini, E. D., Bloch, I. et al. A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes, volume 13, December 2009. doi:10.1016/j.media.2009.07.011

URL http://www.sciencedirect.com/science/article/pii/S136184150900067X

[58] Bilgel, M., Roy, S., Carass, A. et al. Automated anatomical labeling of the cerebral arteries using belief propagation, volume 8669, 2013. ISBN 9780819494436. doi:10.1117/12.2006460

[59] Çabuk, A. D., Alpay, E. and Acar, B. 'Detecting tubular structures via direct vector field singularity characterization'. In 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology. IEEE, August 2010. doi:10.1109/iembs.2010.5628028

[60] Mut, F., Wright, S., Ascoli, G. A. et al. 'Morphometric, geographic, and territorial characterization of brain arterial trees'. *International Journal for Numerical Methods in Biomedical Engineering*, volume 30(7):755–766, January 2014. doi:10.1002/cnm.2627

[61] Beriault, S., Xiao, Y., Collins, D. L. et al. 'Automatic SWI venography segmentation using conditional random fields'. *IEEE Transactions on Medical Imaging*, volume 34(12):2478–2491, December 2015. doi:10.1109/tmi.2015.2442236

[62] Cao, R.-F., Wang, X.-C., Wu, Z.-K. et al. 'A parallel markov cerebrovascular segmentation algorithm based on statistical model'. *Journal of Computer Science and Technology*, volume 31(2):400–416, March 2016. doi:10.1007/s11390-016-1634-6

[63] Danilov, A., Ivanov, Y., Pryamonosov, R. et al. 'Methods of graph network reconstruction in personalized medicine'. *International Journal for Numerical Methods in Biomedical Engineering*, volume 32(8):e02754, November 2015. doi:10.1002/cnm.2754

[64] Maier Oskar, Menze Bjoern H., von der Janina, G. et al. 'ISLES 2015 - a public evaluation benchmark for ischemic stroke lesion segmentation from multispectral MRI'. *Medical Image Analysis*, volume 35:250–269, 2017. ISSN 1361-8415. doi:10.1016/j.media.2016.07.009
URL http://www.sciencedirect.com/science/article/pii/S1361841516301268

[65] Chen Liang, Bentley Paul and Rueckert Daniel. 'Fully automatic acute ischemic lesion segmentation in DWI using convolutional neural networks'. *NeuroImage: Clinical*, volume 15:633–643, 2017. ISSN 2213-1582. doi:10.1016/j.nicl.2017.06.016

URL http://www.sciencedirect.com/science/article/pii/S221315821730147X

[66] Dice Lee R. 'Measures of the amount of ecologic association between species'. *Ecology*, volume 26(3):297–302, 1945. ISSN 00129658, 19399170. doi:10.2307/1932409
URL http://www.jstor.org/stable/1932409

[67] Mah Yee-Haur, Jager Rolf, Kennard Christopher et al. 'A new method for automated high-dimensional lesion segmentation evaluated in vascular injury and applied to the human occipital lobe'. *Cortex; a Journal Devoted to the Study of the Nervous System and Behavior*, volume 56(100):51–63, December 2012. ISSN 0010-9452 1973-8102. doi:10.1016/j.cortex.2012.12.008
URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4071441/

[68] Muda, A. F., Saad, N. M., Waeleh, N. et al. 'Integration of fuzzy C-means with correlation template and active contour for brain lesion segmentation in diffusion-weighted MRI'. In 2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), pp. 268–273, 2015. doi:10.1109/AIMS.2015.88

[69] Tsai, J.-Z., Peng, S.-J., Chen, Y.-W. et al. 'Automatic detection and quantification of acute cerebral infarct by fuzzy clustering and histographic characterization on diffusion weighted MR imaging and apparent diffusion coefficient map'. volume 2014:963032, March 2014

[70] Nielsen, A., Hansen, M. B., Tietze, A. et al. 'Prediction of tissue outcome and assessment of treatment effect in acute ischemic stroke using deep learning'. *Stroke*, volume 49(6):1394–1401, June 2018. doi:10.1161/strokeaha.117.019740

[71] Badrinarayanan, V., Kendall, A. and Cipolla, R. 'SegNet: A deep convolutional encoder-decoder architecture for image segmentation'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 39(12):2481–2495, December 2017. doi:10.1109/tpami.2016.2644615

[72] Matus, S., W., A. G. and Roland, B. 'Real-time diffusion-perfusion mismatch analysis in acute stroke'. *Journal of Magnetic Resonance Imaging*, volume 32(5):1024–1037, 2010. ISSN 1053-1807. doi:10.1002/jmri.22338;10.1002/jmri.22338

[73] He, K., Zhang, X., Ren, S. et al. 'Deep residual learning for image recognition'. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2016. doi:10.1109/cvpr.2016.90

[74] Ranzato, M., Huang, F. J., Boureau, Y. et al. 'Unsupervised learning of invariant feature hierarchies with applications to object recognition'. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2007. ISSN 1063-6919. doi:10.1109/CVPR.2007.383157

[75] Long, J., Shelhamer, E. and Darrell, T. 'Fully convolutional networks for semantic segmentation'. *CoRR*, volume abs/1411.4038, 2014
URL `http://arxiv.org/abs/1411.4038`

[76] Sudre, C. H., Li, W., Vercauteren, T. et al. 'Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations'. *CoRR*, volume abs/1707.03237, 2017
URL `http://arxiv.org/abs/1707.03237`

[77] Ruder, S. 'An overview of gradient descent optimization algorithms'. *CoRR*, volume abs/1609.04747, 2016
URL `http://arxiv.org/abs/1609.04747`

[78] Rumelhart, D. E., Hinton, G. E. and Williams, R. J. 'Learning representations by back-propagating errors'. *Nature*, volume 323(6088):533–536, October 1986. doi:10.1038/323533a0

[79] Kingma, D. and Ba, J. 'Adam: A method for stochastic optimization'. *International Conference on Learning Representations*, December 2014

[80] Rosenblatt, F. 'The perceptron: A probabilistic model for information storage and organization in the brain'. *Psychological Review*, pp. 65–386, 1958

[81] Nwankpa, C., Ijomah, W., Gachagan, A. et al. 'Activation functions: Comparison of trends in practice and research for deep learning', November 2018

[82] Cybenko, G. 'Approximation by superpositions of a sigmoidal function'. *Mathematics of Control, Signals, and Systems*, volume 2(4):303–314, December 1989. doi:10.1007/bf02551274

[83] Hornik, K., Stinchcombe, M. and White, H. 'Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks'. *Neural Networks*, volume 3(5):551–560, January 1990. doi:10.1016/0893-6080(90)90005-6

[84] LeCun, Y., Jackel, L. D., Boser, B. et al. 'Handwritten digit recognition: Applications of neural net chips and automatic learning'. *IEEE Communication*, pp. 41–46, November 1989

[85] Lin, T., Dollár, P., Girshick, R. B. et al. 'Feature pyramid networks for object detection'. *CoRR*, volume abs/1612.03144, 2016
URL `http://arxiv.org/abs/1612.03144`

[86] Krizhevsky, A., Sutskever, I. and Hinton, G. E. 'ImageNet classification with deep convolutional neural networks'. *Communications of the ACM*, volume 60(6):84–90, May 2017. ISSN 0001-0782. doi: 10.1145/3065386
URL `http://doi.acm.org/10.1145/3065386`

[87] Ronneberger, O., Fischer, P. and Brox, T. 'U-Net: Convolutional networks for biomedical image segmentation'. *CoRR*, volume abs/1505.04597, 2015
URL `http://arxiv.org/abs/1505.04597`

[88] Springenberg, J. T., Dosovitskiy, A., Brox, T. et al. 'Striving for simplicity: The all convolutional net'. *CoRR*, volume abs/1412.6806, 2014
URL `http://arxiv.org/abs/1412.6806`

[89] Hochreiter, S. and Schmidhuber, J. 'Long short-term memory'. *Neural Computation*, volume 9(8):1735–1780, November 1997. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735
URL `http://dx.doi.org/10.1162/neco.1997.9.8.1735`

[90] Gers, F. A., Schraudolph, N. N. and Schmidhuber, J. 'Learning precise timing with LSTM recurrent networks'. *J. Mach. Learn. Res.*, volume 3:115–143, March 2003. ISSN 1532-4435. doi:10.1162/153244303768966139
URL `https://doi.org/10.1162/153244303768966139`

[91] Salehinejad, H., Baarbe, J., Sankar, S. et al. 'Recent advances in recurrent neural networks'. *CoRR*, volume abs/1801.01078, 2018
URL `http://arxiv.org/abs/1801.01078`

[92] Schuster, M. and Paliwal, K. K. 'Bidirectional recurrent neural networks'. *IEEE Transactions on Signal Processing*, volume 45(11):2673–2681, November 1997. ISSN 1053-587X. doi:10.1109/78.650093

[93] Shi, X., Chen, Z., Wang, H. et al. 'Convolutional LSTM network: A machine learning approach for precipitation nowcasting'. *CoRR*,

volume abs/1506.04214, 2015
URL `http://arxiv.org/abs/1506.04214`

[94] Gao, Y., Phillips, J. M., Zheng, Y. et al. 'Fully convolutional structured LSTM networks for joint 4D medical image segmentation'. In 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pp. 1104–1108, April 2018. ISSN 1945-8452. doi: 10.1109/ISBI.2018.8363764

[95] Zhang, D., Icke, I., Dogdas, B. et al. 'A multi-level convolutional LSTM model for the segmentation of left ventricle myocardium in infarcted porcine cine MR images'. In 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pp. 470–473, April 2018. ISSN 1945-8452. doi:10.1109/ISBI.2018.8363618

[96] Arbelle, A. and Raviv, T. R. 'Microscopy cell segmentation via convolutional LSTM networks', May 2018

[97] Chen, J., Yang, L., Zhang, Y. et al. 'Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation'. *CoRR*, volume abs/1609.01006, 2016
URL `http://arxiv.org/abs/1609.01006`

[98] Stollenga, M. F., Byeon, W., Liwicki, M. et al. 'Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation'. *CoRR*, volume abs/1506.07452, 2015
URL `http://arxiv.org/abs/1506.07452`

[99] Clevert, D., Unterthiner, T. and Hochreiter, S. 'Fast and accurate deep network learning by exponential linear units (ELUs)'. *CoRR*, volume abs/1511.07289, 2015
URL `http://arxiv.org/abs/1511.07289`

[100] He, K., Zhang, X., Ren, S. et al. 'Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification'. *CoRR*, volume abs/1502.01852, 2015
URL `http://arxiv.org/abs/1502.01852`

[101] Menze, B. H., Jakab, A., Bauer, S. et al. 'The multimodal brain tumor image segmentation benchmark (BRATS)'. *IEEE Transactions on Medical Imaging*, volume 34(10):1993–2024, October 2015. ISSN 0278-0062. doi:10.1109/TMI.2014.2377694

[102] Cicek, O., Abdulkadir, A., Lienkamp, S. et al. '3D U-Net: Learning dense volumetric segmentation from sparse annotation'. In Ourselin, S., Joskowicz, L., Sabuncu, M. R. et al., editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, pp. 424–432. Springer International Publishing, Cham, 2016. ISBN 978-3-319-46723-8

[103] Pierpaoli, C. 'Quantitative brain MRI'. *Topics in Magnetic Resonance Imaging*, volume 21(2), 2010. ISSN 1536-1004
URL `https://journals.lww.com/topicsinmri/Fulltext/2010/04000/Quantitative_Brain_MRI.1.aspx`

[104] Nyul, L. G., Udupa, J. K. and Zhang, X. 'New variants of a method of MRI scale standardization'. *IEEE Transactions on Medical Imaging*, volume 19(2):143–150, February 2000. ISSN 0278-0062. doi:10.1109/42.836373

[105] Candemir, S. and Antani, S. 'A review on lung boundary detection in chest x-rays'. *International Journal of Computer Assisted Radiology and Surgery*, volume 14(4):563–576, February 2019. doi:10.1007/s11548-019-01917-1

[106] Wu, K., Otoo, E. and Shoshani, A. 'Optimizing connected component labeling algorithms'. In Fitzpatrick, J. M. and Reinhardt, J. M., editors, Medical Imaging 2005: Image Processing. SPIE, April 2005. doi:10.1117/12.596105

[107] Zhang, C. and Ma, Y. Ensemble Machine Learning: Methods and Applications. Springer Publishing Company, Incorporated, 2012. ISBN 1441993258, 9781441993250

[108] Kobold, J., Vigneron, V., Maaref, H. et al. 'Stroke thrombus segmentation on SWAN with multi-directional U-Nets'. In IPTA 2019, 2019

[109] Eppelsheimer, M. 'Convolutional neural networks for thrombus segmentation in MRI brain images of stroke patients'. Master's thesis, Universität Regensburg, 2019

[110] Roncoli, A. 'Etude comparative sur les méthodes non supervisées de segmentation des lésions AVC'. Master's thesis, Université Evry Val d'Essonnes, 2019

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Deep Learning for Lesion and Thrombus Segmentation from Cerebral MRI

Keywords: Deep Learning, Segmentation, MRI

Deep learning, the world's best set of methods for identifying objects on images. Stroke, a deadly disease whose treatment requires identifying objects on medical imaging. Sounds like an obvious combination yet it is not trivial to marry the two. Segmenting the lesion from stroke MRI has had some attention in literature but thrombus segmentation is still uncharted area. This work shows that contemporary convolutional neural network architectures cannot reliably identify the thrombus on stroke MRI. Also it is demonstrated why these models don't work on this problem. With this knowledge a recurrent neural network architecture, the logic LSTM, is developed that takes into account the way medical doctors identify the thrombus. Not only this architecture provides the first reliable thrombus identification, it also provides new insights to neural network theory. Especially the methods for increasing the receptive field are enriched with a new parameter free option. And last but not least the logic LSTM also improves the results of lesion segmentation by providing a lesion segmentation with human level performance.

Apprentissage Profond pour la Segmentation de Lésion et de Thrombus dans des IRM Cérébrales

Mots Clés: Apprentissage Profond, Segmentation, IRM

L'apprentissage profond est le meilleur ensemble de méthodes au monde pour identifier des objets sur des images. L'accident vasculaire cérébral est une maladie mortelle dont le traitement nécessite l'identification d'objets par imagerie médicale. Cela semble être une combinaison évidente, mais il n'est pas anodin de joindre les deux. La segmentation de la lésion de l'IRM cérébrale a retenu l'attention des chercheurs, mais la segmentation du thrombus est encore inexplorée. Ce travail montre que les architectures de réseau de neurones convolutionnels contemporaines ne peuvent pas identifier de manière fiable le thrombus sur l'IRM. En outre, il est démontré pourquoi ces modèles ne fonctionnent pas sur ce problème. Fort de cette connaissance, une architecture de réseau neuronal récurrente a été développée, appelée logic-LSTM, capable de prendre en compte la manière dont les médecins identifient le thrombus. Cette architecture fournit non seulement la première identification fiable de thrombus, mais elle fournit également de nouvelles informations sur la théorie des réseaux neuronaux. En particulier, les méthodes d'augmentation du champ récepteur sont enrichies d'une nouvelle option sans paramètre. Enfin, le logic-LSTM améliore également les résultats de la segmentation des lésions en fournissant une segmentation des lésions avec un niveau de performance humaine.

Université Paris-Saclay

Espace Technologique / Immeuble Discovery

Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France