Dissertation

submitted to the
Combined Faculty of Natural Sciences and Mathematics
of the Ruperto Carola University Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Presented by
M.Sc. Fabian Isensee
born in: Aachen, Germany

Oral examination: 24.09.2020

From Manual to Automated Design of Biomedical Semantic Segmentation Methods

Referees: Prof. Dr. Benedikt Brors
PD Dr. Klaus H. Maier-Hein

Acknowledgements

First and foremost I would like to express my gratitude towards my supervisor, Klaus Maier-Hein for his continued guidance and encouragement. His understanding of where a project should move next and what research directions will be relevant in the future are unmatched - it is thanks to his foresight that we took it upon us to develop robust and generalizable segmentation models, which not only constitute a major part of this thesis but have also rewarded us with multiple winning entries in segmentation competitions.

I would also like to thank Benedikt Brors for generously taking me in as an external PhD student and for giving valuable feedback from start to until the completion of this thesis. I would further like to thank Ursula Kummer and Ulrich Schwarz for investing their time as examiners for my upcoming defense.

A special thank you also goes to my clinical partners, in particular Philipp Kickingereder, Gianluca Brugnara, Irada Pflüger and Marianne Schell for entrusting me with their projects and data. Collaborating with you has always been a pleasure, and a fruitful one at that. As a team, we not only eased the transfer of state of the art segmentation models into clinical practice but also published the corresponding methods in high ranking journals.

A particularly warm thank you goes to my good friend and colleague Paul F. Jaeger, with whom I have collaborated closely on two of the projects presented in this thesis. His technical expertise, knack for principled evaluations and exceptional talent for comprehensively condensing complex methods were pivotal for the success of these projects.

Speaking of colleagues, one simply cannot leave out the drone racing, mezcal drinking, whiskey tasting, nerf gun shooting alpha madness that was 'deep fridge'. It is an honor to have worked among the best of the best: Paul F. Jaeger (again!), Sebastian Wirkert, Jens Petersen, Simon Kohl, Peter Full, Gregor Köhler, Jakob Wasserthal and Tobias Ross. Despite the never-ending shenanigans, this was (surprisingly) by far the most

productive and entertaining time during my PhD, underlining how quintessential an inspiring work atmosphere is!

I would like to thank Dasha Trofimova, Jens Petersen and Gregor Köhler for proof reading parts of this thesis.

I would also like to express my gratitude towards my parents, Karin and Christian Isensee for their ongoing support, advice and grounding. Your insights and life lessons were invaluable!

Last but by absolutely no means least, I am greatly indebted to my family. My loving wife Katharina has been nothing but supportive, particularly in those last months of my PhD where over hours and work dominated weekends grew increasingly frequent. While this has put a lot of stress onto you as well, you have continuously ensured that I could focus on my work and finish my thesis. Without your occasional 'kick in the butt', I would probably not have finished it for another couple of months. I would also like to mention my lovely daughter Maike, who, even though she may be too young to realize it, continuously puts a smile on my face. Kids grow up so quickly and it is simply amazing to see her make so much progress in such a short time. I am lucky to have the two of you and am looking forward to spending more time with you in the coming months.

Abstract (English)

Digital imaging plays an increasingly important role in clinical practice. With the number of images that are routinely acquired on the rise, the number of experts devoted to analyzing them is by far not increasing as rapidly. This alarming disparity calls for automated image analysis methods to ease the burden on the experts and prevent a degradation of the quality of care. Semantic segmentation plays a central role in extracting clinically relevant information from images, either all by themselves or as part of more elaborate pipelines, and constitutes one of the most active fields of research in medical image analysis. Thereby, the diversity of datasets is mirrored by an equally diverse number of segmentation methods, each being optimized for the datasets they are addressing. The resulting diversity of methods does not come without downsides: The specialized nature of these segmentation methods causes a dataset dependency which makes them unable to be transferred to other segmentation problems. Not only does this result in issues with out-of-the-box applicability, but it also adversely affects future method development: Improvements over baselines that are demonstrated on one dataset rarely transfer to another, testifying a lack of reproducibility and causing a frustrating literature landscape in which it is difficult to discern veritable and long lasting methodological advances from noise.

We study three different segmentation tasks in depth with the goal of understanding what makes a good segmentation model and which of the recently proposed methods are truly required to obtain competitive segmentation performance. To this end, we design state of the art segmentation models for brain tumor segmentation, cardiac substructure segmentation and kidney and kidney tumor segmentation. Each of our methods is evaluated in the context of international competitions, ensuring objective performance comparison with other methods. We obtained the third place in BraTS 2017, the second place in BraTS 2018, the first place in ACDC and the first place in the highly competitive KiTS challenge. Our analysis of the four segmentation methods reveals that competitive segmentation performance for all of these tasks can be achieved with a standard, but well-tuned U-Net architecture, which is surprising given the recent focus in the literature on finding better network architectures. Furthermore,

we identify certain similarities between our segmentation pipelines and notice that their dissimilarities merely reflect well-structured adaptations in response to certain dataset properties. This leads to the hypothesis that we can identify a direct relation between the properties of a dataset and the design choices that lead to a good segmentation model for it.

Based on this hypothesis we develop nnU-Net, the first method that breaks the dataset dependency of traditional segmentation methods. Traditional segmentation methods must be developed by experts, going through an iterative trial-and-error process until they have identified a good segmentation pipeline for a given dataset. This process ultimately results in a fixed pipeline configuration which may be incompatible with other datasets, requiring extensive re-optimization. In contrast, nnU-Net makes use of a generalizing method template that is dynamically and automatically adapted to each dataset it is applied to. This is achieved by condensing domain knowledge about the design of segmentation methods into inductive biases. Specifically, we identify certain pipeline hyperparameters that do not need to be adapted and for which a good default value can be set for all datasets (called *blueprint parameters*). They are complemented with a comprehensible set of heuristic rules, which explicitly encode how the segmentation pipeline and the network architecture that is used along with it must be adapted for each dataset (*inferred parameters*). Finally, a limited number of design choices is determined through empirical evaluation (*empirical parameters*). Following the analysis of our previously designed specialized pipelines, the basic network architecture type used is the standard U-Net, coining the name of our method: nnU-Net ("No New Net"). We apply nnU-Net to 19 diverse datasets originating from segmentation competitions in the biomedical domain. Despite being applied without manual intervention, nnU-Net sets a new state of the art in 29 out of the 49 different segmentation tasks encountered in these datasets. This is remarkable considering that nnU-Net competed against specialized manually tuned algorithms on each of them. nnU-Net is the first out-of-the-box tool that makes state of the art semantic segmentation methods accessible to non-experts. As a framework, it catalyzes future method development: new design concepts can be implemented into nnU-Net and leverage its dynamic nature to be evaluated across a wide variety of datasets without the need for manual re-tuning.

In conclusion, the thesis presented here exposed critical weaknesses in the current way of segmentation method development. The dataset dependency of segmentation methods impedes scientific progress by confining researchers to a subset of datasets available in the domain, causing noisy evaluation and in turn a literature landscape in which results are difficult to reproduce and true methodological advances are difficult to discern. Additionally, non-experts were barred access to state of the art segmentation for their custom datasets because method development is a time consuming trial-and-

error process that needs expertise to be done correctly. We propose to address this situation with nnU-Net, a segmentation method that automatically and dynamically adapts itself to arbitrary datasets, not only making out-of-the-box segmentation available for everyone but also enabling more robust decision making in the development of segmentation methods by enabling easy and convenient evaluation across multiple datasets.

Abstract (German)

Die digitale Bildgebung spielt in der klinischen Praxis eine immer wichtigere Rolle. Obwohl die Zahl der routinemäßig aufgenommenen Bilder stetig zunimmt, steigt die Zahl der für die Bildanalyse zuständigen Experten bei weitem nicht so schnell an. Diese alarmierende Ungleichheit erfordert automatisierte Bildanalysemethoden, um die Experten zu entlasten und eine Verschlechterung der Versorgungsqualität zu verhindern. Semantische Segmentierung spielt eine zentrale Rolle bei der Extraktion klinisch relevanter Informationen aus Bildern, entweder isoliert betrachtet oder als Teil komplexerer Pipelines, und stellt eines der aktivsten Forschungsfelder der medizinischen Bildanalyse dar. Dabei spiegelt sich die Vielfalt der Datensätze in einer ebenso vielfältigen Anzahl von Segmentierungsmethoden wider, die jeweils für die von ihnen adressierten Datensätze optimiert sind. Die daraus resultierende Methodenvielfalt ist nicht ohne Nachteile: Die Spezialisierung dieser Methoden führt zu einer Datensatzabhängigkeit, die es unmöglich macht, sie ohne weitere Optimierung auf andere Segmentierungsprobleme zu übertragen. Dies führt nicht nur zu Problemen bei der Anwendbarkeit, sondern wirkt sich auch nachteilig auf die zukünftige Methodenentwicklung aus: Verbesserungen gegenüber Baselines, die an einem Datensatz demonstriert werden, lassen sich nur selten auf einen anderen übertragen, was zu einer mangelnden Reproduzierbarkeit und damit zu einer frustrierenden Literaturlandschaft führt, in der es schwierig ist, fundamentale und zukunftsweisende methodische Fortschritte vom Rauschen zu unterscheiden.

Wir untersuchen drei verschiedene Segmentierungsprobleme mit dem Ziel zu verstehen, was ein gutes Segmentierungsmodell tatsächlich ausmacht und welche der kürzlich vorgeschlagenen Methoden wirklich erforderlich sind, um eine kompetitive Segmentierungsgenauigkeit zu erzielen. Zu diesem Zweck entwerfen wir state-of-the-art Segmentierungsmodelle für die Segmentierung von Hirntumoren, kardialen Substrukturen sowie Nieren und Nierentumoren. Um einen objektiven Leistungsvergleich mit anderen Methoden zu gewährleisten wird jede unserer Methoden im Rahmen internationaler Wettbewerbe bewertet. Hierbei haben wir den dritten Platz in BraTS 2017, den zweiten Platz in BraTS 2018, den ersten Platz in ACDC und den ersten Platz im hochkompe-

titiven KiTS Wettbewerb erhalten. Unsere Analyse der vier Segmentierungsmethoden zeigt, dass eine kompetitive Segmentierungsqualität für all diese Aufgaben mit einer standardmäßigen, aber gut eingestellten U-Net Architektur erzielt werden kann, was angesichts des jüngsten Fokus in der Literatur auf der Suche nach besseren Netzwerkarchitekturen überraschend scheint. Darüber hinaus stellen wir bestimmte Ähnlichkeiten zwischen unseren Segmentierungspipelines fest und lernen, dass ihre Unterschiede lediglich gut strukturierte Anpassungen als Reaktion auf bestimmte Datensatzeigenschaften widerspiegeln. Dies führt zu der Hypothese, dass wir eine direkte Beziehung zwischen den Eigenschaften eines Datensatzes und den Designentscheidungen identifizieren können, die zu einem guten Segmentierungsmodell für diesen Datensatz führen.

Basierend auf dieser Hypothese entwickeln wir nnU-Net, die erste Methode, die die Datensatzabhängigkeit traditioneller Segmentierungsmethoden überwindet. Traditionelle Segmentierungsmethoden müssen von Experten entwickelt werden, die einen iterativen Trial-and-Error-Prozess durchlaufen, bis sie eine gute Segmentierungspipeline für einen bestimmten Datensatz identifiziert haben. Dieser Prozess führt letztendlich zu einer festen Pipeline-Konfiguration, die möglicherweise mit anderen Datensätzen inkompatibel ist und eine umfangreiche Neuoptimierung erfordert. Im Gegensatz dazu verwendet nnU-Net eine generalisierende Methodenvorlage, die dynamisch und automatisch an jeden neuen Datensatz angepasst wird. Dies wird durch die Kondensation von Domänenwissen über das Design von Segmentierungsmethoden in Form von induktivem Bias erreicht. Insbesondere identifizieren wir bestimmte Pipeline-Hyperparameter, die nicht angepasst werden müssen und für die ein guter Standardwert für alle Datensätze eingestellt werden kann (sogenannte *Blueprint Parameter*). Sie werden durch einen verständlichen Satz heuristischer Regeln ergänzt, die explizit kodieren, wie die Segmentierungs-Pipeline und die zugehörige Netzwerkarchitektur abhaengig von den Datensatzeigenschaften angepasst werden muessen (*inferierte Parameter*). Schließlich wird eine begrenzte Anzahl von Designentscheidungen durch empirische Evaluation bestimmt (*Empirische Parameter*). Motiviert durch die Analyse unserer zuvor entworfenen spezialisierten Pipelines wird als grundlegender Netzarchitekturtyp das Standard-U-Net verwendet, das den Namen unserer Methode prägt: nnU-Net ("No New Net"). Wir wenden nnU-Net auf 19 verschiedene Datensätze an, die aus Segmentierungswettbewerben im biomedizinischen Bereich stammen. Obwohl nnU-Net ohne manuellen Eingriff angewendet wird, erreicht es bei 29 der 49 verschiedenen Segmentierungsaufgaben, die in diesen Datensätzen vorkommen, einen neuen Bestwert. Dies ist bemerkenswert, wenn man bedenkt, dass nnU-Net bei jedem Datensatz gegen spezialisierte, manuell angepasste Algorithmen konkurriert. nnU-Net ist das erste sofort einsatzbereite Tool, das modernste semantische Segmentierungsmethoden auch für Laien zugänglich macht. Als Framework katalysiert es die zukünftige Methodenentwicklung: Neue Designkonzepte können in nnU-Net implementiert werden und seine dynamische Charakteristik

nutzen, um über eine Vielzahl von Datensätzen ausgewertet zu werden, ohne dass eine manuelle Neuabstimmung erforderlich ist.

Zusammenfassend lässt sich festhalten, dass die hier vorgestellte Dissertation kritische Schwächen in der derzeitigen Art und Weise der Methodenentwicklung zur Segmentierung verdeutlicht. Die Datensatzabhängigkeit der Segmentierungsmethode behindert den wissenschaftlichen Fortschritt, indem sie die Forscher auf eine Teilmenge der in der Domäne verfügbaren Datensätze beschränkt, was zu einer verrauschten Auswertung und damit zu einer Literaturlandschaft führt, in der die Ergebnisse nur schwer reproduzierbar und echte methodische Fortschritte nur schwer zu erkennen sind. Darüber hinaus wurde Laien der Zugang zu einer Segmentierung nach dem Stand der Technik für ihre individuellen Datensätze bisher verwehrt, weil die Methodenentwicklung bisher ein zeitaufwändiger Trial-and-Error-Prozess war, der Fachwissen erforderte, um korrekt durchgeführt werden zu können. Um dieser Problematik zu begegnen, schlagen wir nnU-Net vor, eine Segmentierungsmethode, die sich automatisch und dynamisch an beliebige Datensätze anpasst und nicht nur als Segmentierungsmethode für jeden verfügbar ist, sondern auch eine robustere Entscheidungsfindung bei der Entwicklung von neuen Segmentierungsmethoden vereinfacht, indem sie eine einfache und bequeme Auswertung über mehrere Datensätze hinweg ermöglicht.

Contents

1. Introduction

Gathering information about the innards of the human body is quintessential for modern medicine. Only when we know what is going on inside can we judge what disease a patient may be suffering from or whether the therapy they are getting has the desired effect. One way of achieving this goal could be to open up the patient in a surgical intervention. While physical access to the affected parts of the body certainly opens up the valuable opportunity to do a visual inspection as well as take biological samples, it also comes with obvious adverse effects to the patient's health and well-being.

Imaging techniques, such as Magnetic Resonance Imaging (MRI) or Computed Tomography (CT), on the other hand offer the possibility of examining the patient's body without causing physical harm. They constitute a particularly powerful tool in modern medicine, because they not only allow the visualization of tissue properties, but do so in a spatially resolved way enabling the analysis of potential heterogeneities of the disease. Images are regularly used for planning surgical interventions and radiotherapy as they provide insights into the outline and surrounding of the target structure [1]. The lack of adverse effects, especially with MRI, furthermore opens up the time axis: Whereas surgical interventions and biopsies can not be done at arbitrary time points due to their invasive nature, images can be acquired as often as necessary making them the perfect tool for monitoring diseases, such as tumors, over time. Modern imaging techniques hereby offer unprecedented flexibility: Acquisition time can be traded for spatial resolution, with long acquisitions enabling spatial resolutions down to 100 microns [2], whereas, on the other end of the spectrum, a lower desired resolution results in acquisition times fast enough to monitor the heart as it beats [3, 4].

Given their inherent benefits, it is unsurprising that medical images are on the rise. The Clinical radiology UK workforce consensus report 2018 [5] notes that the number

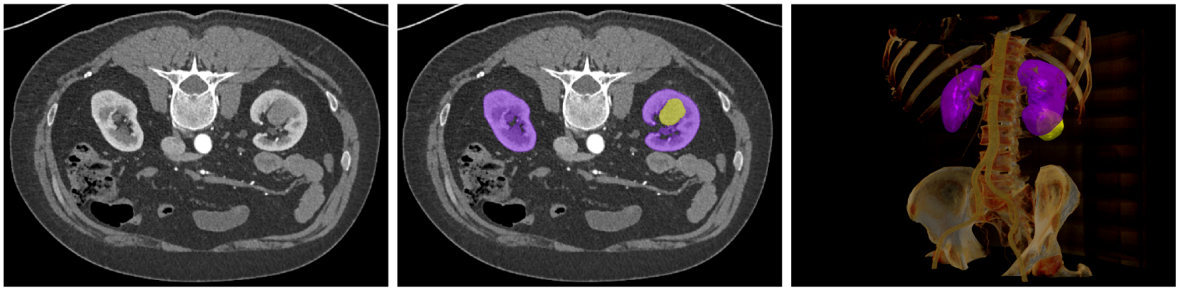

Figure 1.1.: **Example for semantic segmentation in Medical Images.** Left: axial slice of an abdominal CT image. Middle: Overlay of the raw image with a segmentation map. Purple encodes kidney and yellow encodes tumor. Right: Volume rendering of the patient to highlight the 3D nature of the segmentation problem. Image is taken from the KiTS [15] test set and the segmentation is generated by our automatic segmentation method presented in Section 3.3.

of CT and MRI acquisitions has increased by 48% and 54%, respectively, between 2012 and 2018. At the same time, the number of radiologists has increased by only 21%, resulting in a severe lack of experts for image interpretation with potentially detrimental effects to the quality of healthcare. This causes "delayed diagnosis of cancer and critical findings" and leaves the clinics "unable to provide a safe and reliable radiology service" [5]. Interpretation of medical images is a complex task, and as such requires concentration and time to do correctly. Independent scientific studies have already confirmed that spending less time per scan increases the error rate by as much as 17% [6].

With more images being acquired in clinical practice on the one hand and a lack of radiologists on the other, the question arises how the quality of healthcare can be maintained or maybe even improved in the future. Fueled by recent advances in computer vision [7, 8, 9] as well as recently published methods for medical image analysis achieving or even surpassing radiologist-level performance [10, 11, 12], one possible answer to this question is automation. Not only can automation take away tedious repetitive work from the radiologists, freeing them up to deal with more pressing matters, but it also has the potential to increase the quality of care. Automated methods are fast to compute, easy to scale and yield reproducible results. They furthermore take away the human component and thus address issues that naturally arise from it: substantial variations in skill, large inter-rater variability [13] and inattentive blindness [14].

This thesis focuses on automated image processing algorithms for semantic segmentation. In semantic segmentation, all voxels in an image are assigned a class label indicating what type of object it belongs to. A typical example for semantic segmentation in the medical domain is provided in Figure 1.1. It shows kidney and kidney

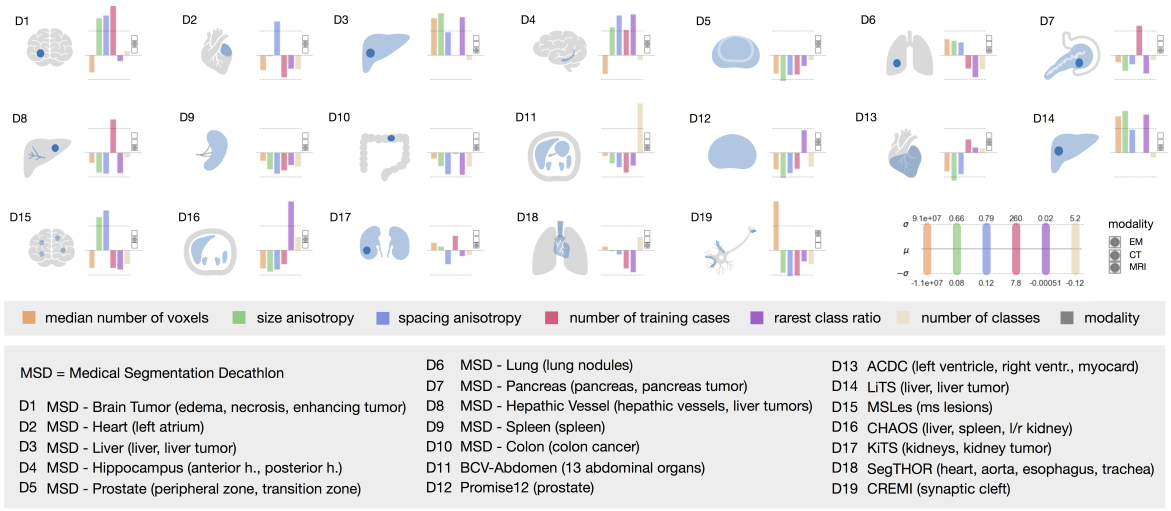

Figure 1.2.: **Dataset diversity in Medial Image Segmentation.** Each dataset comes with its own unique set of properties and peculiarities that need to be considered when designing segmentation methods for them. As a result, methods developed on one (type of) dataset are often incompatible with others, requiring constant redesign and tuning of methods when confronted with a new segmentation problem. This dataset dependency of methods severely hampers progress in the field and makes segmentation inaccessible to non-experts. Figure reproduced from [23].

tumor segmentation in abdominal CT images with a raw axial slice shown to the left, an overlay with the segmentation in the middle and a 3D volume rendering to the right highlighting the three dimensional nature of the problem.

The development of automated segmentation models is one of the most researched areas in medical image computing [16] and has numerous applications. In radiotherapy, radiologists spend a substantial amount of time manually delineating organs at risk as well as the cancerous region(s) so that subsequent irradiation can be planned to hit the target structure with the maximum intensity while sparing the most important organs. Automating this process frees up valuable time for the radiologist [1]. In the diagnosis and therapy response assessment of tumors, limited time for image annotation causes the substitution of accurate manual delineations with less accurate measurements [17, 18, 19]. Here, automatic segmentation not only saves time but also increases the accuracy and reliability of the measurements [10] to the benefit of the patient. Segmentations are also an essential part of many image processing pipelines. They serve for example as intermediary representation for the diagnosis of retinal diseases [12] or for decoding molecular properties of tumors via radiomics [20, 21, 22].

Due to the diversity and flexibility of imaging techniques, the diversity of datasets

in the medical domain is exceptional. Figure 1.2 documents this by extracting key properties that are relevant for designing appropriate segmentation methods. Each of the 19 dataset comes with its own peculiarities, requiring an algorithm to be tediously adapted to it. As a result, modern semantic segmentation algorithms, which are almost exclusively based on deep convolutional neural networks, are highly specific to the dataset and application they are designed for, dealing with class imbalance [24, 25], heterogeneous [26] and anisotropic [4, 27] voxel spacing, large variability in image sizes [28, 29, 30], ambiguities [31, 32, 33], potential errors in the reference segmentations [34] and many more. The design of such methods requires appropriate training data, time and expertise. Facilitated by the availability of high quality publicly available training data, current research is mostly focused around only a couple of different segmentation problems with the most prominent examples being abdominal organ segmentation [35, 15, 36, 37], brain lesions [38, 39, 40], heart [29, 4, 41] and prostate [42, 29].

With image properties being diverse, and corresponding algorithms requiring careful tuning and optimization to optimally handle them, existing segmentation methods are not compatible with datasets other than the one they were designed for. For each new, unique dataset, the current state of the art dictates that experts design suitable segmentation methods, spending a lot of time in the form of numerous trial and error experiments to find a good configuration. This issue not only hampers methods from being available out of the box for a broad range of datasets but also causes severe issues when researching new segmentation models, especially if said research is not done on one of the standardized datasets. In particular, the dataset incompatibility issue of segmentation methods is also present in baseline methods, such as the famous U-Net [43, 44] architecture, causing the lack of a guaranteed high quality baseline to compare new methods against. Instead, authors have to reimplement baselines themselves so that they match the requirements of their dataset, a process that often evokes suboptimal hyperparameter settings and low baseline performance, making it difficult to gauge whether the conclusions drawn in the corresponding publications can be relied upon.

1.1. Contributions

With semantic segmentation methods currently being bound to the dataset they were developed on, both the research of new methods as well as their application to arbitrary new datasets are severely impeded. The objective of this thesis is to break the dataset dependency of segmentation models by developing a framework that takes a basic segmentation algorithm, here based on the famous U-Net architecture [43], and makes it generalizable in the sense that this model is automatically adapted to and can then be trained on any dataset in the domain.

In order to achieve this goal, we first look into three case studies, namely brain tumor segmentation (Section 3.1), cardiac segmentation (Section 3.2) and kidney and kidney tumor segmentation (Section 3.3). For each use case, we manually design and tune separate segmentation pipelines. We experiment with different network architectures, method configurations as well as pre-and postprocessing techniques with the goal of understanding what makes or breaks a state of the art segmentation model on each of these tasks. We furthermore will gain insights into why methods cannot readily be transferred between datasets. All our developments are tested in the context of segmentation challenges to ensure an objective performance evaluation on standardized datasets.

In **Manual Design of Segmentation Pipelines** (Chapter 3) we make the following contributions:

- Brain tumor segmentation in multi-modal MRI is considered to be one of the most difficult problems in medical image segmentation [38]. The class imbalance, the amorphous nature of the tumors as well as the potentially limited contrast between tumor and healthy tissue are only three of the many challenges it poses. We develop two different segmentation methods to tackle this task: First, we develop a model based on a 3D U-Net with a residual encoder. We show that this model produces radiologist-level accuracy on a large multi-institutional cohort comprising more than 2000 MRI acquisitions. Furthermore, we evaluate the model on the BraTS 2017 challenge [45] where it obtained the third place. We then developed a second model intended to push the limits of a baseline architecture, the standard 3D U-Net. It uses region-based training, tailored postprocessing and an optimized loss function to specifically target the difficulties of brain tumor segmentation. It is evaluated on the BraTS 2018 challenge, where it obtained the second place out of 64 participating teams [45].
- Cardiac substructure segmentation in cine-MRI suffers from anisotropic data, slice-misalignments as well as imaging artifacts. We show how these difficulties can be overcome by developing a method based on ensembling standard 2D and 3D U-Nets. The method is evaluated on the Automatic Cardiac Diagnosis Challenge [4] where it obtained the first place.
- Kidney and kidney tumor segmentation in large abdominal CT scans poses problems with balancing the receptive field of the network with the target spacing for resampling. Furthermore, tumors are difficult to discern from cysts and can be hard to detect in the first place because they have very heterogeneous texture. We use this segmentation problem to study the differences between the standard 3D U-Net and several variants thereof which use residual connections in the encoder. Our best model is then evaluated in the Kidney and Kidney Tumor segmentation challenge

[15, 28] where it obtained the first place out of 105 participants.
- Finally we provide a thorough analysis of the different segmentation problems and the key aspects of our methods that allowed them to be successful. Specifically, we attempt to connect dataset properties to suitable design choices which could serve as best practices for finding good segmentation models on new, previously unseen datasets.

We then set out to break the dataset dependency of segmentation methods by developing a framework that automatically configures new models to arbitrary datasets. The core idea behind this framework is to automatically determine the defining properties of a dataset and how a model must be designed to deal with them effectively. To achieve this goal, we use the experience gathered from the case studies in the previous chapter.

Specifically, in Chapter **Automatic Design of Segmentation Pipelines** (Chapter 4), we make the following contributions:

- We revisit the key aspects that made our models in the previous chapter successful. We use them to formulate guiding principles on how segmentation methods could be adapted to yield good results on a new dataset with unique properties.
- For the first time, we formalize the relationship between dataset properties and method configuration required for a successful model. The implementation of this formalization yields nnU-Net, a framework for automated generation of segmentation methods.
- We demonstrate the effectiveness of this approach by participating in 10 highly competitive segmentation challenges comprising 19 different datasets and 49 segmentation tasks. Without manual intervention, our method sets a new state of the art on the majority of segmentation tasks even though it competes against manually tuned algorithms on each of the datasets. Remarkably, many recently proposed segmentation methods use sophisticated network architectures whereas our results were achieved by adapting the standard U-Net architecture, coining the name of our framework: nnU-Net ("No new net").
- nnU-Net has far reaching consequences for medical image segmentation:
  - nnU-net is the fist segmentation algorithm that can be used out of the box on arbitrary datasets and still deliver state of the art segmentation accuracy. As such, it constitutes a valuable tool for researchers and clinicians who need semantic segmentation both for their research as well as clinical applications. nnU-Net requires no expert knowledge to run and does not require excessive GPU resources, making it accessible to a broad audience.
  - For the first time a single algorithm can be used on any dataset in the domain. This is particularly valuable in a research environment where methodological con-

tributions are often demonstrated on new datasets for which no optimized baseline implementations exist. nnU-Net is such an optimized baseline and comparing against it will yield more convincing evidence when proposing new methods.
– As a framework, nnU-Net catalyzes research in medical image segmentation. nnU-Net is modular on one hand, allowing for easy integration of new ideas, while being flexible on the other, enabling researchers to evaluate their method on a large number of datasets. Evaluation across multiple datasets results in substantially more reliable conclusions while also preventing overfitting.

1.2. Outline

The outline of this thesis is as follows. In chapter 2 we revisit the state of the art in medical image segmentation. We furthermore look into issues associated with the current way of developing segmentation, particularly those related to the dataset dependency of segmentation methods and the need for proper hyperparameter tuning. Then, in chapter 3 we develop four new state of the art methods for three different segmentation problems. These methods are then analysed and used as a basis for developing nnU-Net (chapter 4), a framework that automatically generates state of the art segmentation models for arbitrary datasets. We close with a discussion in chapter 5 on how this framework could transform the way we do method development in medical image segmentation.

2. Related Work

This chapter lays the foundations for the methodological innovations presented in chapters 3 and 4. We first give a brief overview of modern pre deep learning methods for medical image segmentation, followed by a journey through the history of deep learning based methods as well as a presentation of the current state of the art. We close this chapter with a discussion about the current way research is being done in the domain and the shortcomings associated with it.

2.1. Pre deep learning era

2.1.1. Shape-based methods

Statistical shape models have been researched extensively in the past [46], in particular for the segmentation of organs and for shape analysis [47, 48]. For segmentation, they use a surface representation of the object of interest that is fitted to the image information. There are several ways of representing the surfaces, with the most common one being based on landmarks [49]. Landmarks are specific locations on the shape of the objects. It is quintessential to assign the same location within the shape to the same landmark on all shapes found in the training set in a process called correspondence optimization. As a preprocessing step, shapes are aligned and normalized using these corresponding landmarks. The underlying principle of statistical shape models is to model the distribution of shapes (and appearances) encountered in a training set to get a shape (and appearance) prior that can then be fitted to a new image. In the most simple case, a shape prior can be found by running (kernel) principal component analysis on the normalized and aligned training shapes. This will capture the most relevant modes of variation while suppressing potential noise in the data [50, 51, 52].

Appearance priors can be built by analyzing the intensity profile perpendicular to the landmark positions [53, 54], region-based appearance modelling [55] or by incorporating additional non-local information around the landmarks [56]. When applying statistical shape models to a new, unknown image, the shape must either be initialized close to the desired target structure or special measures must be taken to make the algorithm robust to random initialization [57]. Then, in an iterative process, the underlying model parameters are adapted to fit the image information while also adhering to the learned prior distribution of shape and appearance. For a more complete overview of statistical shape models, please refer to [46].

Statistical shape models have the inherent advantage that they are strongly constrained by the shapes encountered in the training cases. This allows them to only produce plausible shapes, making them robust even if the number of training cases is low. Their robustness, however, can also cause them to be not as precise in cases where the shape of the object in the image cannot be represented by their parametrization (bias-variance trade-off). While shape models are a good fit for segmenting structures that follow a certain shape and appearance pattern, such as organs, they are less well suited for the segmentation of amorphous and heterogeneous structures such as tumors or vessels. Furthermore, shape models must be retrained for each new target structure to be segmented.

2.1.2. Atlas-based methods

Atlas-based segmentation methods basically treat a segmentation problem as a registration problem [58]. Single atlas segmentation requires only one manually delineated example image. To generate a segmentation for an unknown target image, one of the images is registered to the other, meaning that it is rigidly transformed (rotation, scale, shearing, translation) and elastically deformed until the two images match. After registration the segmentation of the atlas can be transferred to the target image. Using a single atlas is, however, often insufficient for capturing the broad anatomical variation and can result in inaccurate segmentations [59]. Multi-atlas-segmentation (MAS) [60, 61] makes use of multiple atlases to improve upon this deficiency. Hereby, the target image is registered pairwise with each of the available atlases. After transferring all segmentations to the target image, the final segmentation can then be obtained via label fusion (this can be majority voting in the most simple case). The quality of multi-atlas segmentation is determined mostly by the quality of the registration and the strategy applied for label fusion. We refer to [62] for a more detailed overview of atlas-based medical image segmentation.

Atlas-based methods are data efficient, with single-atlas based methods yielding acceptable results with only a single training case. With the exception of modern data-driven

algorithms for label fusion, new target structures can be added to the atlas and thus transferred to novel images without the need to adapt or retrain the method. Similarly to shape models, atlas-based segmentation methods are well suited for segmenting structures that follow a specific pattern in both their shape and location within the body. They fall short, however, in the segmentation of pathologies and highly irregular shapes. Furthermore, the lack of explicit shape and appearance modeling can result in unrealistic segmentations, for example caused by registration errors. Finally, offloading segmentation to registration brings several issues with it. First and foremost, registration in itself is a difficult problem that is an active area of research by itself [63, 64, 65] and has not yet been solved to perfection. Furthermore, registration is computationally expensive and often results in high run time, in particular in multi-atlas segmentation where pairwise registration to all atlases yields a large number of registrations that need to be done.

2.1.3. Pixel-based Methods

Pixel-based methods rely on a classifier to make a decision for each pixel in the image independently. The features available to the classifier are hereby crucial for the success of this approach. To ensure that decisions can be made under consideration of both local as well as more global information, it is upon the researcher to design the features appropriately. Ideally, features encode a large variety of image properties across multiple scales, such as the presence or absence of edges, texture information or smoothed intensity information. Popular feature extractors are edge detectors [66], Haar wavelets [67], intensity gradients and texture features or even simple Gaussian smoothing filters [68, 69]. Features are computed from the original image. Their output is a new, transformed image of identical shape, referred to as feature representation. For each voxel, a feature descriptor can be collected by accumulating the values found at the pixel location across all feature maps. Since most features also encode information about the surroundings of the voxel they belong to, spatial information is encoded implicitly in these representations, allowing the classifier to incorporate the surrounding context into its decisions even though it only operates on a per-pixel level. To prevent the classifier from being overwhelmed by a large number of features, and to select features that are appropriate for the task at hand, feature selection algorithms can be used to identify the most relevant representations [70]. The most popular type of classifier used for pixel classification is the Random Forest [71]. It excels through its balance of expressivity, robustness to overfitting, ability to handle a large number of features effectively as well as its low computational complexity. Successful applications are prevalent in tumor segmentation [72] where the local texture and intensity information is particularly predictive. Due to the pixel-wise decision of the classifiers, postprocessing techniques are often applied to smooth the resulting segmentation and

enforce spatial consistency, for example through the use of guided filters [73], graph cut [74] or conditional random fields [75].

Pixel-based methods excel at segmentation tasks where the structure of interest has irregular shapes and can be found with more local rather than image-global information, in particular whenever it is identifiable by its texture and intensity profile. The vast pool of possible features, classifiers and postprocessing methods makes them extremely versatile, but also requires researchers to be experienced with all the aspects of the pipeline to be successful. Pixel-based methods also struggle with incorporating global information into the decision process because they require the design of features that can encode it. This task becomes increasingly difficult the larger the receptive field of the features needs to be. Finally, depending on the choice and number of features, computation times for feature extraction can be cumbersome.

2.2. Deep Learning-based Methods

2.2.1. Image Classification with Convolutional Neural Networks

The success of convolutional neural networks (CNN) started when the AlexNet architecture [7] won the ImageNet image classification [76] challenge in 2012 by a large margin. Since then, the state of the art not only in image classification but also image segmentation [77] object detection [78] and instance segmentation [79] have been dominated by this type of methods.

CNNs differ from the previously presented methods in that they do not require any prior information, image registration or manual feature design. They are entirely data driven and learn directly from training data. Building appropriate network architectures is hereby key for enabling the learning process: the network architecture is in a certain way a template that can be molded during training to extract the necessary information directly from the image. The extraction of information is hereby handled by stacking convolutional layers with nonlinearities in between. The convolutions act similarly to the handcrafted features used in the pixel-based methods presented in Section 2.1.3 by transforming their input to generate new feature representations. The quintessential difference to the previously described features (many of which can also be expressed by convolutions) is that the kernel weights of the transformation are not set by the researcher but instead treated as learnable parameters during training. Another critical aspect for the success of CNNs is the stacking of these transformations. Only the very first convolution operates on the raw image values. Every successive layer then takes the feature representation of the previous layer as input, enabling the network to recombine previously computed representations and thereby learning increasingly expressive features with each layer.

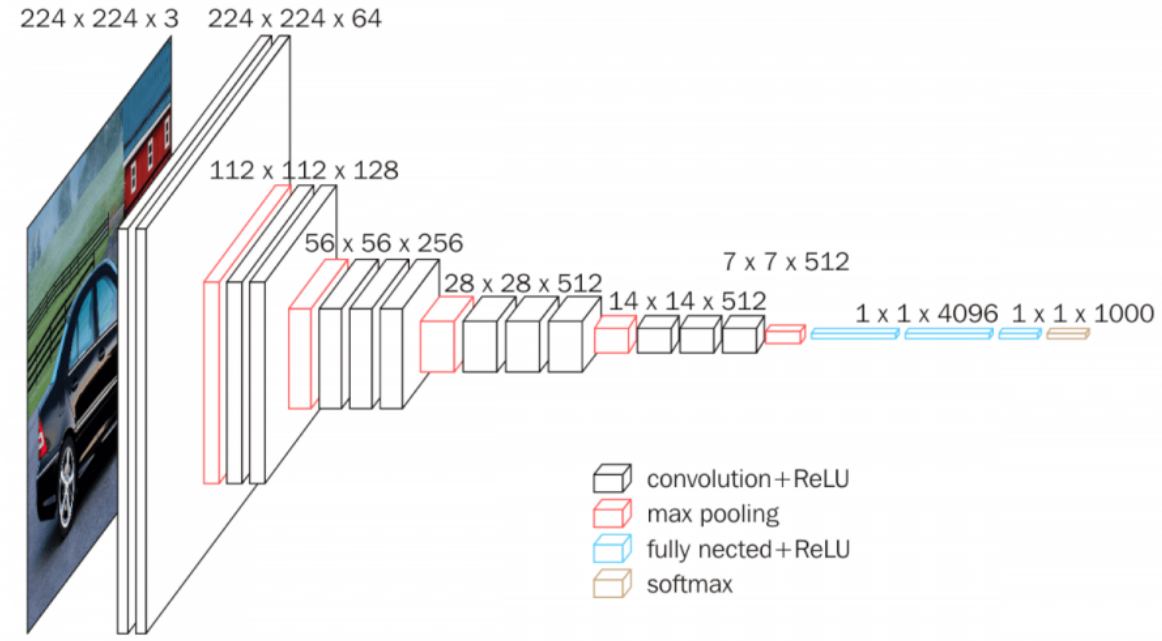

Figure 2.1.: **VGG16 network architecture.** Convolutional neural networks for image classification extract feature representations by stacking convolutional layers. Hereby, each convolution takes the feature representations of the previous block as input, allowing the network to recombine existing representations into new, more expressive ones. After a set number of convolutions, max pooling layers aggregate the feature representation spatially and reduce the size of the feature maps in half, allowing the following convolutions to operate on what is effectively a larger area of the image and thereby aggregating more global information. The pattern of alternating feature extraction (via convolutions) and spatial aggregation (pooling) is repeated until the features have a small spatial extent but contain image global information. Then, fully connected layers are used to do the final classification. Each black rectangular block represents the output of a convolution with the associated tensor size written next to it. Red blocks represent max pooling operations. Blue boxes are fully connected layers. Figure taken from [80].

The way most modern CNNs for image classification are constructed is by alternating computational blocks for feature extraction with spatial aggregation. One prominent example, the VGG-16 network [81], is shown in Figure 2.1. After some number of convolutional layers, the size of the feature representation is reduced with a max pooling operation. This operation only retains the highest value found within a certain region in the input feature map (here 2×2). Since it is applied at a coarser grid (here every other pixel), the resulting feature representation has half the spatial extent than the feature map it was applied to. Utilizing pooling operations increases the receptive field of the convolutions because, at a fixed kernel size and a lower resolution feature maps, they effectively operate on a larger proportion of the image. Alternating convolutions with pooling operations enables the network to successively transform local information, such as the presence or absence of edges, corners or textures into global representations which finally enable the classification of the entire image. It is common practice to increase the number of convolutional kernels (and with it the number of feature representation) as the spatial resolution decreases to increase the representational power of the model.

CNNs are purely data driven and require a large number of training cases. The famous ImageNet challenge [76], for example, comprises one million images with 1000 different classes. Training is most commonly done by stochastic gradient descent: A small subset of the training database, called a minibatch, is passed through the network (forward pass) and the networks classification output is compared with the ground truth information. A loss function hereby serves as a metric for how good the network output is. The most commonly used loss function for image classification is the categorical cross-entropy. In the subsequent backwards pass, the gradients of the loss with respect to all parameters in the network (these are usually the kernel parameters of the convolutions) are obtained with backpropagation [82]. Hereby, the gradients are computed starting at the last layer and propagated through the layers in the network in reverse order by applying the chain rule. Finally, all model parameters are updated by subtracting their gradient multiplied by some constant (the so called learning rate). Finding a good set of hyperparameters for training CNNs is quintessential for obtaining good performance. Due to relatively long training times, codependency of hyperparameters and a large number thereof, finding a good setting is considered difficult and requires careful optimization, either through expertise, grid search or AutoML-like approaches [83]. Parameters that must be optimized, apart from the network architecture, include the minibatch size, learning rate, momentum term, input size, kernel parameter initialization and many more.

Modern state of the art classification algorithms still follow the basic scheme of alternating feature computation and spatial pooling, but improve upon the way the

representations are computed. He et al. [9, 84] observed that deeper networks do not always improve the results. According to the authors, this result is counter-intuitive because the solution space of the shallower networks is a mere subset of their deeper counterparts. They explain this shortcoming by the inability of convolutions to model the identity function, which should allow them to bypass not needed feature computations. They propose to offload the extraction of feature representations into so-called *residual blocks*, a stream that branches off the main network and adds its result (residual) back to the input feature maps. The resulting architectures are called Resnets and have been shown to enable the construction of substantially deeper networks while improving the accuracy in the process. GoogleNet [85, 8] computes representations in each step not by using a single convolution or residual block, but instead splits the feature computation into several streams, each with a reduced number of resulting representations. The representations of the streams are concatenated before being passed to the next step. The rationale between the multiple streams is to increase the diversity of the operations used at once (for example convolutions with different kernel sizes), thus making the feature extraction process more flexible. Densenets [86] are specifically optimized for network depth. Instead of adding the result of a feature computation block to its input (as done in Resnets), they concatenate it. This ultimately results in a substantially improved gradient flow, because layers are densely connected and gradients can be passed from the tail of the network all the way to the front with no steps in between.

2.2.2. Fully Convolutional Models (FCN)

The first segmentation algorithms based on CNNs used the same architecture as classification networks, but instead of classifying the entire image they were trained to predict the semantic class of the center pixel of their input [87, 88]. The network input was hereby often significantly smaller than the typical image size to prevent excessive padding at the image borders. To predict an entire image, these networks needed to be slided across the whole image, pixel by pixel, to generate a complete segmentation map. This approach is computationally inefficient because feature representations computed in one forward pass cannot be reused for another, resulting in very long run times.

This issue was recognized by Long, Shelhamer et al [89, 90] who designed FCN, the first architecture for fully convolutional image segmentation. The key idea behind fully convolutional architectures is to utilize only operations whose parameters are independent of the input size (such as convolutions, hence the name), thus allowing them to be applied to arbitrary image sizes. This enables the re-use of computed feature maps and makes the approach computationally efficient. At the core of their approach they used standard Imagenet pretrained networks [7, 85, 81]. This reduces the number

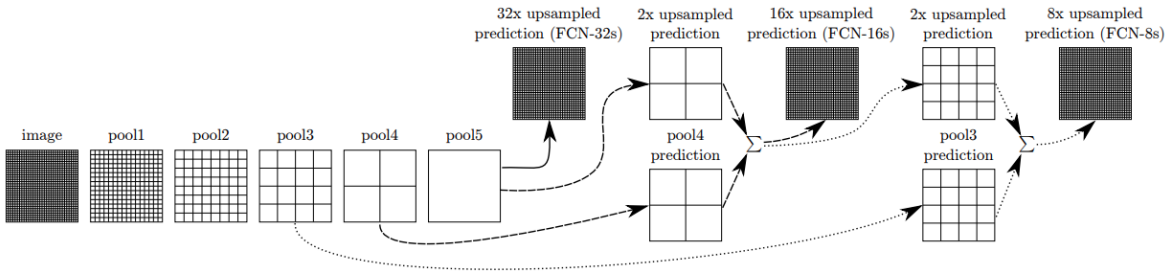

Figure 2.2.: **FCN Network.** A standard Imagenet pretrained classification network is converted into a fully convolutional CNN by replacing its fully connected layers with 1×1 convolutions and dropping the final pooling layer. Left: The varying resolutions of Alexnet are represented by the corresponding pooling layers (see [7]). Right: Reconstruction of full image resolution segmentations by convolution transposed. FCN-32s utilizes a segmentation head located in pool5 to obtain a segmentation at 1/32 image resolution, which is then upscaled with convolution transposed. FCN-16s and FCN-8s use two and three segmentation heads, respectively, which are located also at higher resolution feature maps (pool4 and pool3). They are recombined after upsampling through summation. Figure taken from [89].

of training cases required for training because the kernel weights of the convolutions are already trained to produce meaningful representations instead of initialized randomly. To convert the networks to be fully convolutional, the final fully connected layers are converted to 1x1 convolutions and the pooling operation preceding these layers is dropped. One major drawback of this approach is the low resolution of the segmentation output. As mentioned previously, CNNs need to successively reduce the spatial extent of their representations to enable the convolutional kernels to *see* large proportions of the image simultaneously, which is a requirement for correctly identifying large objects. Translating this pattern to the type of information being available to the network at a given layer, there is a lot of spatial information and little semantic information in the early layers and a lot of semantic information but little spatial information at the final layers. The networks used by Long et al. make use of five pooling operations, which results in coarse segmentation outputs that are downsampled by a factor of 32 (2^5) with respect to the original input ('output stride 32').

To improve the output resolution they use a convolution transposed at the end of the network which upsamples the segmentations back to the original image resolution. A convolution transposed effectively constitutes a learned upsampling that incorporates class-specific prior information into the process, thus increasing the fidelity of full resolution segmentations over bilinear upsampling. Still, the resulting segmentations are quite coarse and cannot capture fine structure in the image. To further improve the situation, they experimented with adding additional segmentation heads at two finer

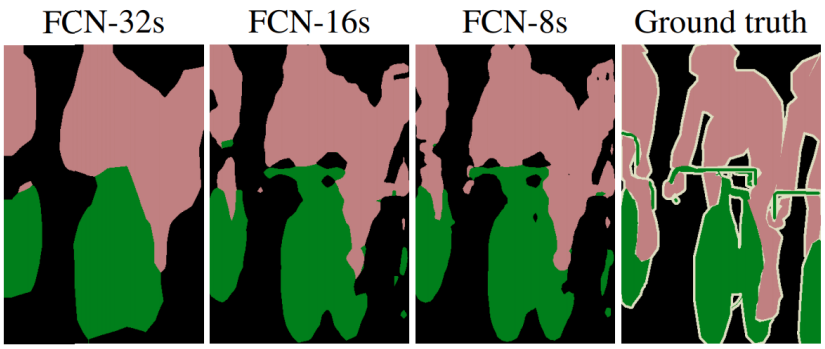

Figure 2.3.: **Impact of output stride on segmentation fidelity.** FCN-32s generates a segmentation at 1/32 image resolution (output stride 32), FCN-16s at 1/16 and FCN-8s at 1/8. Resolution and fidelity of the generated segmentation increase the smaller the output stride. Figure taken from [89].

locations (at stride 16 and 8) and combining the upsampled outputs of the heads by addition (see Figure 2.2). Weights for the segmentation heads as well as the convolution transposed were fine-tuned end-to-end on PascalVOC [91]. Figure 2.3 shows how utilizing additional predictions from higher resolution layers gives finer details in the segmentations.

2.2.3. U-Net based methods

In order to produce a precise segmentation, a network needs to extract both, what objects are in the image (semantic information) as well as exactly which pixels belong to them (spatial information). As we have seen previously, CNNs need to reduce spatial information in order to obtain more semantic information. Thus, once the semantic information is available, the spatial information must somehow be recovered. Long et al. [89] solved this by upsampling the low resolution segmentations with a convolution transposed. They furthermore generated segmentations at different resolution outputs and merged them together. While this worked reasonably well, the downside of this approach is that only the stride 32 output has access to the full semantic information while the other segmentation heads have more spatial information at their disposal but potentially limited knowledge about the semantic of the pixels. This is problematic because the segmentations generated at earlier layers cannot gain access to the semantic information required for them to be accurate. The lack of proper recombination of spatial and semantic information severely limits the accuracy FCN can achieve. It is also the main reason why this architecture cannot create segmentations at output stride 1, and thus always requires upsampling.

These shortcomings were addressed in the famous U-Net architecture [43]. It constitutes a significant improvement over FCN [89], both in terms of how spatial and

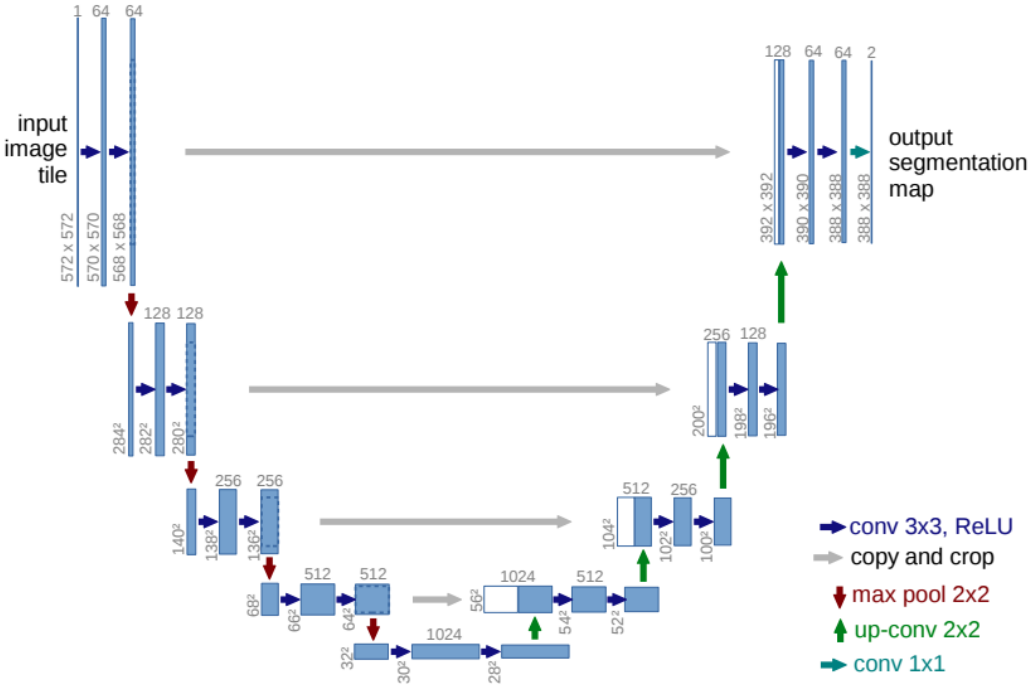

Figure 2.4.: **U-Net architecture.** U-Net consists of three major components: the encoder (left), the decoder (right) and skip connections connecting them. The encoder follows the same pattern as image classification networks: convolution and pooling operations are alternated, generating increasingly expressive representations at the cost of a reduced spatial resolution. In order to generate high fidelity segmentation maps at image-level resolution, the semantic information present at the bottleneck is then successively reconstructed in the decoder network. Hereby, increasingly high resolution feature maps stemming from the encoder (via skip connections) are concatenated to the upsampled feature maps. This enables the network to take the semantic information from the encoder and precisely localize it in the decoder until the original image resolution is obtained. Only then the final segmentation is generated. Figure taken from [43].

semantic information are recombined in the network as well as the resolution of the output segmentations. U-Net, which is shown in Figure 2.4, consists of three major components: The *encoder* is similar to FCN and follows the well-known structure of image classification networks. It alternates feature computation and spatial aggregation, thus increasing the semantics in its representations while losing spatial information. Where U-Net differs from previous approaches is in the reconstruction of the spatial information after the end of the encoder. The simple segmentation heads that were used by FCN are replaced by a whole *decoder* network which effectively mirrors the architecture of the encoder and reverses the downsampling of the former step by step. By itself, the decoder would not be able to generate high resolution segmentations because all the spatial information needed for that would need to pass through the bottleneck and some of it would be lost. To alleviate this problem, *skip connections* forward feature representations from the encoder. They are concatenated with the upsampled feature maps in the encoder and then used jointly in the following convolutions. Due to its unique architecture, U-Net elegantly recombines local and global information at several stages throughout the network. This allows it to successively broadcast the semantic information gathered by the encoder to the original image size, resulting in segmentations generated at output stride 1. This property is particularly interesting for medical image segmentation, where the exact boundary position is quintessential, for example for radiotherapy planning [1].

2.2.4. Medical Image Segmentation Beyond the U-Net

In recent years, many improvements upon the original U-Net architecture have been proposed. The 3D U-Net [44] transitioned the original architecture to use 3D convolutions to better deal with the 3D nature of biomedical images. V-Net [25] also uses 3D convolutions but additionally replaces the feature computation blocks with residual layers [84] for improved gradient flow and representational power. They were also among the first [24, 25] to replace the most commonly used cross-validation or weighted cross-validation [92] loss with the Dice loss to directly optimize the metric used to evaluate segmentations. Design patterns that were found to be effective in image classification on the ImageNet database were also quickly transferred into a U-Net-like segmentation architecture. [93] for example make use of densely connected convolutional layers, a principle that was also adapted in the medical domain [94]. [95] augmented the U-Net with squeeze and excitation modules [96] and introduced their own variant thereof. Others added attention modules to the U-Net [97] to improve the localization for organs.

Still today in 2020, U-Net architectures and their derivatives define the state of the art in medical image segmentation [98, 45, 35, 4, 99].

2.3. AutoML

Automatic machine learning (AutoML), as the name implies, seeks to take away the human component in designing and configuring machine learning models by replacing as many steps during method development as possible with automation [100]. It targets machine learning (ML) experts and non-experts alike: "ML experts can profit from AutoML by automating tedious tasks like hyperparameter optimization (HPO) leading to a higher efficiency. Domain experts can be enabled to build ML pipelines on their own without having to rely on a data scientist" [101]. There are many different ways of introducing automation into model design, with hyperparameter optimization, model selection, feature design and neural architecture search being the most common ones.

Hyperparameter optimization can be a tedious and time consuming task which, depending on the classification algorithm used, can also take a lot of compute resources to be done successfully. Yet, it is a quintessential step in obtaining good machine learning models: proper hyperparameters can often not be set a priori as different datasets may require different parameters to yield optimal results [102]. Grid search [103, 104] is one of the most straightforward ways of addressing this problem: a plausible value range for each hyperparameter is provided by the experimenter, along with a corresponding coarseness with which is should be evaluated. Then, each possible combination is tested and the best result is returned. Given appropriate ranges, this approach yields good results, but at a high computational cost. The high cost combined with the restricted values hyperparameters can attain (due to the grid) is often problematic in practise. Recent evidence even suggests that random search [105] should be preferred over grid search as it finds better configurations with less computational overhead. Bayesian optimization tackles the problem from a different angle by using "an algorithm to build a probability model of the objective function, and then uses this model to select the most promising hyperparameters" [106]. This allows the algorithm to probe only promising hyperparameter combinations and prevents the unnecessary exploration of combinations that are unlikely to give good results. Genetic algorithms [107] approach this problem from yet another angle by constructing a population of hyperparameter sets and using evolutionary approaches to successively increase the fitness of the population.

Each classification problem has different characteristics and different machine learning models may be more or less suited to address it [102]. Model selection in the context of AutoML refers to the automated testing of different models and automatically selecting the best based on some validation score. To ensure a good selection, model selection is often done in conjunction with hyperparameter optimization, for example in the popular auto-sklearn framework [83].

The performance of a machine learning model, in particular those that use non-deep

learning methods (for example the methods described in Section 2.1.3) strongly depends on the availability of a suitable feature set [102]. There are some approaches that make an attempt at automated feature design (e.g. [108]), but a large performance gain can also be achieved by simply extracting as many features as possible and then running feature selection [70, 109, 110, 20] to cut down on the number of unnecessary features.

Recently, the area of AutoML that has certainly received the most attention is neural architecture search [111, 112, 113]. Instead of using manually designed sequences of convolutional layers, such as the ones we presented above (Section 2.2), this area of research focuses on how these architectures can be derived automatically. Hereby, very different approaches can be selected, for example based on evolutionary strategies [114] or fully differentiable search spaces [111]. While early methods required immense compute resources to be run effectively, more recent methods [115] specifically attempt to cut down on the computational complexity. There already exist initial attempts at making neural architecture search viable for medical image segmentation [116, 117] but these so far fall short of simpler, manually designed network architectures. We refer to [106] for a more extensive overview of recent advances in neural architecture search.

2.4. Competitions in Medical Image Segmentation

The medical image analysis community is extraordinarily active in developing new segmentation methods to cope with the many diverse datasets that can be encountered in the domain. In this context, a large number of competitions (also referred to as *challenges*) has been conceived with the goal of either encouraging the development of methods for a dataset that is yet unsolved or for providing a standardized environment in which algorithms can be tested and evidence for methodological improvements can be derived. Many of these challenges are held in conjunction with the Conference on Medical Image Computing and Computer Assisted Interventions, the largest conference in the domain.

The general structure of a competition is as follows: A fixed number of training cases is released to the public containing both the original images as well as the corresponding segmentations which, ideally, were generated by medical experts. These are then used by the participants to develop and train their models. Test images (without their segmentations) are either also released to the participants or participants need to submit their algorithm to be evaluated by the challenge organizers. Evaluation is done by comparing the segmentation maps generated by the participating algorithms with the withheld reference segmentation. Finally, the metrics used for comparison are aggregated and a challenge ranking is created.

Metrics used for evaluation can be grouped in two major groups: overlap and distance-based metrics. Depending on the segmentation task, other metrics may also be used.

The by far most popular metric for evaluating segmentations in the medical domain is the Dice coefficient [118, 119]. It measures how well two segmentation maps overlap. The Dice coefficient (also often called the Dice score or simply Dice for brevity) is defined as:

$$\text{DICE}(A, B) = \frac{2A \cap B}{|A| + |B|} \tag{2.1}$$

Where A and B are the two segmentation maps to be compared. The Dice coefficient is computed individually for each class present in the image. $A \cap B$ measures the number of pixels with which they overlap. $|A|$ and $|B|$ denote the number of pixels in map A and B, respectively. A perfect overlap results in a Dice score of 1, no overlap in a Dice score of 0. If both A and B do not contain a class, the respective Dice score is undefined (this special case receives special treatment in some challenges [38], also see Section 3.1.3).

When it comes to distance-based metrics, the Hausdorff distance (HD) is the most popular. In the context of segmentations, it measures the maximum distance between the two surfaces of A and B. It is again computed for each class individually. Perfect agreement in the segmentations results in a HD of 0, disagreement causes increasingly high HD the further away from the reference the segmentation is. Due to its sensitivity to outliers (a single false positive far away results in a huge Hausdorff distance), challenges often opt to use the HD95 metric, which takes the 95th percentile of the surface distances instead of their maximum.

There are multiple ways of aggregating metrics to a challenge rank with the most commonly used being metric aggregation (for example by averaging) followed by the actual ranking. The discussion of the different ranking schemes is beyond this thesis. A comprehensive discussion and overview are provided in [16].

2.5. Discussion

There exists an enormous body of literature addressing semantic segmentation of medical images. It is complemented by numerous competitions, some of which exceed 700 submissions to their leaderboards [1]. Although there are varying beliefs about what exactly constitutes a good segmentation algorithm, there is a unilateral consensus that variants of the U-Net (i.e. encoder-decoder with skip connections) are state of the art for supervised semantic segmentation problems. Over the years, numerous improvements over the vanilla U-Net [43] and its 3D counterpart [44] have been proposed to

[1] `https://kits19.grand-challenge.org/evaluation/results/`

push the state of the art, with most of them revolving around elaborate architectural modifications.

In an ideal world, the effectiveness of new methods would be demonstrated by applying them to as many datasets as possible thereby either exceeding state of the art performance on multiple competitions or, at the least, improving upon appropriate standardized baseline implementations. The real world is, unfortunately, quite far away from this reality.

More often than not, methodological improvements and the associated claims are demonstrated on only a single dataset. And even if multiple datasets are used, they are often too similar (such as both being abdominal CT scans) thus limiting the generality of the claims. Especially in the medical domain where datasets often contain only in the order of one to several hundred training cases and about half as many test cases, the inherent noisiness of the results as well as the potential for overfitting raise the question whether a general methodological improvement demonstrated on one (type of) dataset will actually translate to other segmentation problems or, in the extreme case, even hold up to a different random seed.

What further complicates the situation is that proposed methods are often not evaluated in the context of competitions, thus severely hampering an objective assessment of their performance. Instead, authors revert to taking some popular model as their baseline, such as the 3D U-Net [44] or the V-Net [25], and demonstrate improved performance relative to them. This strategy is, however, severely flawed. In the Introduction (Chapter 1, in particular Figure 1.2) we have touched on the dataset diversity in the medical domain and the need for dataset-specific adaptations that goes along with it. This translates to models that were developed on some dataset to be incompatible with the dataset properties of another. These restrictions naturally apply to the baselines as well: With the 3D U-Net being developed for *Xenopus* kidney segmentation [44] and the V-Net being developed for prostate segmentation [25] they simply cannot be taken as they are and applied to arbitrary datasets in the domain. As a consequence, authors need to reimplement their baselines and retune their hyperparameters to match the dataset(s) they are working with, a process that is not standardized and error prone, ultimately resulting in unreliable and potentially underperforming baselines. In particular, hyperparameters are sometimes tediously adapted to the proposed method whereas tuning is mostly disregarded for the baseline, for which authors sometimes simply use a copy of the hyperparameters used for their proposed method. As a result, the baseline method may not perform at its best, suggesting an improvement when in reality there exists a different set of hyperparameters for which the baseline by far exceeds the proposed solution.

To underline the impact of hyperparameter tuning, Figure 2.5 presents our analysis of

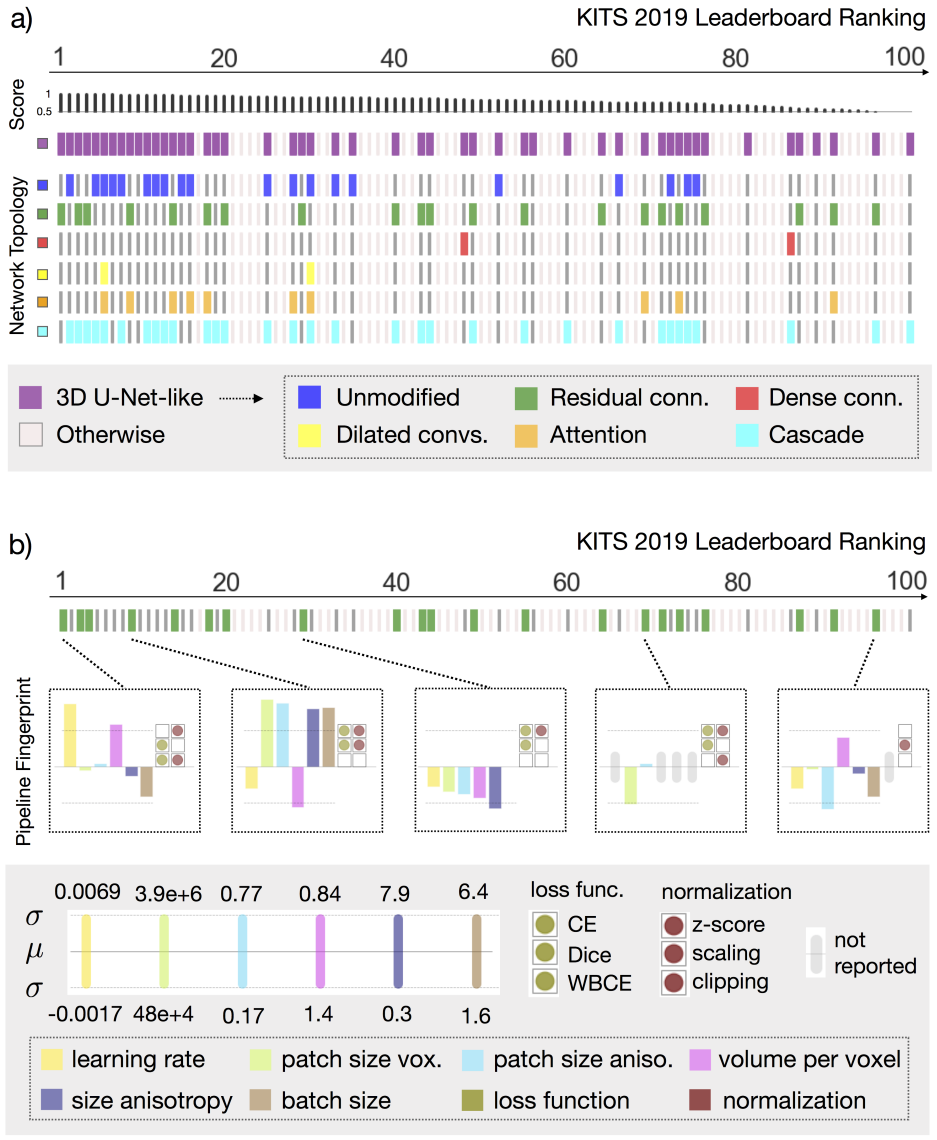

Figure 2.5.: **Hyperparameter tuning of deep learning-based segmentation methods.** Analysis of the 100 submissions to the Kidney and Kidney Tumor Segmentation Challenge 2019. a) Coarse categorization of leaderboard entries by architecture variation. All top 15 methods were 3D U-Net-like, i.e. used encoder-decoder style architectures with skip connections, 3D convolutions and output stride 1. No clear pattern about which variant consistently outperforms the others can be derived. Furthermore, none of the variants constitutes a necessary requirement for good performance. b) Analysis of all methods similar to the challenge-winning entry (non-cascaded 3D U-Net like architecture with residual connections). The methods vary drastically in their performance as well as their selected hyperparameters. No clear connection between hyperparameters and model performance can be established, highlighting the difficulty of hyperparameter optimization. Figure reproduced from [23].

the Kidney and Kidney Tumor Segmentation (KiTS) challenge, the largest competition at MICCAI in 2019 with 100 leaderboard entries. As can be seen in a), all methods in the top 15 were offspring of the 3D U-Net architecture from 2016, confirming its impact on the field of medical image segmentation. When dissecting the 3D U-Net models further into their architectural variants, we do not observe a clear pattern favoring one of the variants over the others. In fact, none of the recently introduced architectural modifications (e.g. residual connections [84, 25], dense connections [86, 93], attention mechanisms [97], or dilated convolutions [77, 34]) seem to represent a necessary condition for good performance. This contradicts the accompanying publications, where for example a plain U-Net was shown to be outperformed by a U-Net with attention gates [97].

Interestingly, each architectural variant can be found all across the leaderboard indicating that the selection of proper hyperparameters may have a substantially larger impact on model performance than the exact model architecture. Identifying a good set of hyperparameters is a difficult and complex process in which co-dependencies of parameters as well as dataset-specific peculiarities need to be considered. To get a glimpse at this problem in the context of the KiTS challenge, we analyzed all entries that use the same architectural variant as the challenge-winning contribution, a non-cascaded 3D U-Net with residual connections. In Figure 2.5 b), each of these models is represented by its key configuration parameters illustrating design choices made by the authors. There appears to be no clear trend linking the choice of parameters to model performance, underlining the complexity of hyperparameter optimization in deep learning methods. This observation stands in stark contrast with the reporting in the literature which almost exclusively focuses on newly introduced network architectures and often disregards the selection of hyperparameters and the process of how they were obtained. Considering the variability in model performance and the large impact of the hyperparameters, this analysis raises questions about the validity of utilizing non-standardized and manually re-tuned baseline when proposing methodological improvements.

With this in mind, research in medical image segmentation is overshadowed by a literature landscape in which even experts struggle to ascertain which methods really constitute a veritable and long lasting improvement over baselines. Disregarding the importance of hyperparameter optimization, especially regarding the baseline method is a major problem not only in this research area but has also been observed in other communities as well [120, 121]. This is particularly important in the medical domain where the dataset diversity causes a coupling of methods to the one (type of) dataset they were developed for and a constant need for manually retuning hyperparameters when applied to other datasets. Not only does this prevent the standardized applica-

tion of baseline algorithms, thus ensuring researchers can objectively measure potential improvements, but it also causes proposed methods to be applicable only to a narrow band within the segmentation problems posed by the domain. As highlighted in Section 1.1, this thesis will demonstrate how the dataset dependency of segmentation methods can be broken by automatically determining appropriate method configurations for each individual datasets. This method not only outperforms the current state of the art on the majority of datasets but also addresses multiple issues in the field: it can be used as high-quality standardized baseline, as framework for future model development or simply as an out-of-the-box tool making medical image segmentation available to non-experts.

3. Manual Design of Segmentation Pipelines

In this chapter we will conduct three case studies of semantic segmentation segmentation problems in the medical domain. We will look at brain tumor segmentation in section 3.1, heart segmentation in section 3.2 and kidney and kidney tumor segmentation in section 3.3 . For each of these problems we will develop state of the art segmentation methods which are evaluated on respective challenge datasets. In preparation for the next chapter, we will then discuss the choices made in each of these algorithms in order to determine which design principles may generally be related to a good segmentation performance.

3.1. Brain Tumor Segmentation

3.1.1. Motivation

Gliomas are the most frequent type of primary brain tumors in adults. Prognosis is poor, with patients suffering from the more aggressive high grade gliomas having a median survival rate of only two years or less [122]. Due to the severity of these tumors, treatment options are often drastic and entail surgical removal of the affected tissue as well as chemotherapy, immunotherapy and radiation therapy. Magnetic resonance imaging (MRI) techniques are widely used throughout the clinical pipeline, from diagnosis and (potential) surgery planning all the way to monitoring treatment success over time. Systematic analysis of the images reveals crucial characteristics of the tumors, such as the presence or absence of areas that accumulate Gadolinium contrast agent as well as the overall size of the tumor. Quantitative measurements are particularly important for assessing treatment success in the form of *progression free survival*, a measure that is increasingly often considered as an endpoint in clinical trials [17]. Response Assessment in Neuro-Oncology (RANO) [17] is the state of the art for measuring therapy response with MRI in both clinical practice [123] and clinical trials [124]. To obtain an estimate of the tumor size, RANO requires clinicians to identify the axial slice with the largest visible contrast-enhancing tumor and to draw a set of perpendicular diameters measuring the spatial extent of this tumor region. This process is repeated for each individual lesion and the total tumor burden is then estimated as the sum of products of the perpendicular diameters [17].

While this approach has certainly been designed with practical considerations in mind, it has obvious drawbacks that substantially impact its accuracy and reliability. First, by using axial slices only, it relies on the assumption that tumors grow in a spherical shape. This is, however, often inaccurate and larger tumor sizes may be observed in coronal or sagittal slices instead. This is particularly problematic when considering that tumor growth is substantially influenced by the surrounding anatomy of the brain and may also be affected by treatment-related effects such as large necrotic regions or

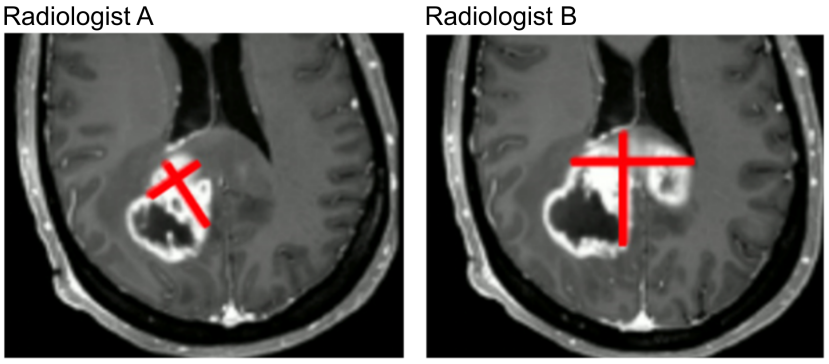

Figure 3.1.: **Inter-rater variability in diameter-based tumor burden estimation.** Tumor burden estimation based on RANO [17] requires the radiologist to identify the axial slice with the largest tumor size followed by estimating the tumor size with perpendicular diameters. This two-step manual approach introduces significant leeway for subjectivity and ultimately results in a high inter-rater variability with potentially harmful consequences.

surgical scars [125, 126]. Furthermore, measurements based on perpendicular diameters in a manually selected axial slice are highly subjective, causing large inter-rater variability [10] (see Figure 3.1) that may have therapeutic consequences for the patients. It is therefore unsurprising that volumetric assessment of tumor volume has been a recent focus with several studies attesting it superior accuracy and reliability in therapy response assessment over the perpendicular diameters within a single axial slice as used by RANO [127, 128]. However, due to the amount of time required to generate manual segmentation of the images, volumetric measurements lack practicability for clinical settings.

Robust automated methods for volumetric tumor measurements can therefore have significant impact on clinical workflows as they combine the best of both worlds: requiring no manual interaction frees up valuable time of the clinician to focus on more urging aspects of patient care whereas the volumetric and automatic nature of the measurements ensure high accuracy and reproducibility. Development of such an algorithm is, however, not a straightforward task as brain tumor segmentation is certainly one of the most difficult tasks in medical image analysis due to the inherent challenges associated with it [38]: tumors can be recognized in the images by slight intensity and texture changes relative to their surroundings. They grow amorphously and exhibit no clear patterns in size, shape and location. Furthermore, their growth can deform the surrounding brain, thus reducing the amount of prior information that could be used to detect them. Figure 3.2 shows a typical example for the complex shape of high grade gliomas. The fine structures of the enhancing tumor region (green) as well as the unclear borders of the necrosis and non-enhancing tumor regions underline the

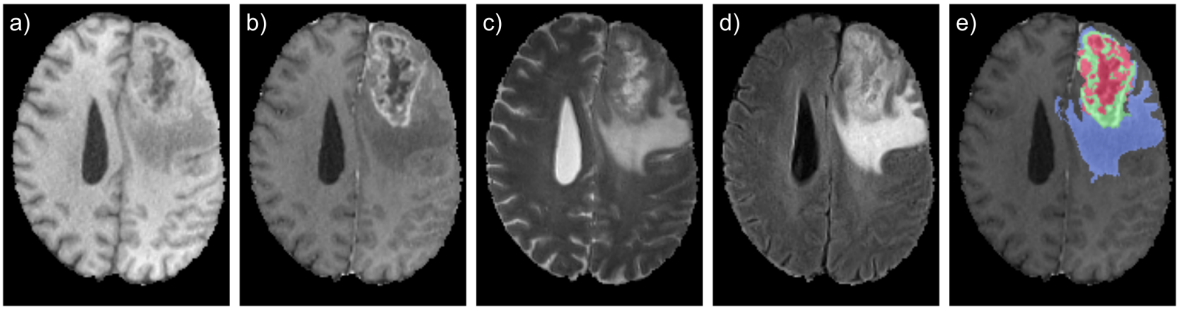

Figure 3.2.: **MRI of a high grade glioma.** a) T1w, b) T1c (T1w with contrast agent), c) T2w, d) Flair and e) Manual segmentation of the tumor compartments overlaid with the T1c image. The edema is shown in blue, the enhancing tumor region in green and the necrotic core as well as non-enhancing tumor regions in red. The image shown originates from the BraTS 2018 challenge (case *CBICA_ABE_1*).

difficulty and ambiguity associated with annotating brain tumors in MRI.

The Brain Tumor Segmentation Challenge (BraTS) [38] is an annual competition that provides a large training dataset (335 cases as of 2019) and catalyzes the development of brain tumor segmentation methods. Deep learning methods in particular have recently been dominating the competition [99, 129, 98, 130, 92, 34] underlining the potential for these types of methods.

3.1.2. Automated Tumor Response Assessment with Artificial Neural Networks

This section is based on the following publications ([10] and [130]):

Kickingereder, P.*, **Isensee, F.***, Tursunova, I., Petersen, J., Neuberger, U., Bonekamp, D., Brugnara, G., Schell, M., Kessler, T., Foltyn, M., Harting, I., Sahm, F., Prager, M., Nowosielski, M., Wick, A., Nolden, M., Radbruch, A., Debus, J., Schlemmer, H.-P., Heiland, S., Platten, M., von Deimling, A., van den Bent, M. J., Gorila, T., Wick, W., Bendszus, M. & Maier-Hein, K. H. (2019). Automated quantitative tumour response assessment of MRI in neuro-oncology with artificial neural networks: a multicentre, retrospective study. The Lancet Oncology, 20(5), pp.728-740. `https://doi.org/10.1016/S1470-2045(19)30098-1`

Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., & Maier-Hein, K. H. (2017). Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge. In International MICCAI Brainlesion Workshop (pp. 287-297). Springer, Cham.

(*: shared first authorship)

Note that this chapter explicitly omits extensive details regarding the collection of cases for the HD-train, HD-test and EORTC datasets, their annotation procedure and the clinical evaluation of the method because these parts of the publication [10] were contributed by my co-author, Philipp Kickingereder. The clinical integration of the resulting segmentation method was done in collaboration with my co-author Jens Petersen. The conception and implementation of the segmentation method, the presented experiments, the (non-clinical) evaluation and the application to the BraTS 2017 challenge were contributed by me.

3.1.2.1. Introduction

Despite the popularity and size of the BraTS dataset, the algorithms developed in the context of the competition have so far not been proven to be sufficiently robust for transfer into clinical practice. One of the major reasons for this is the lack of patients with treatment-induced alterations of the brain, such as resection cavities and surgical scars in this dataset.

In the following section we describe our fully automated brain tumor segmentation algorithm for volumetric tumor progression analysis. We show that this algorithm is highly accurate and robust and thus qualifies for application in clinical practice. Integration of our segmentation pipeline into clinical infrastructure ensures seamless embedding into existing workflows and delivers maximum usability. Our segmentation algorithm is currently being used in the Heidelberg University Hospital.

3.1.2.2. Dataset description

A large, diverse and accurately annotated training dataset is as important to the success of a model as model itself. To ensure that our model is able to cope with the large variability that occurs in the shape, localization and appearance of tumors, a suitable training set is required. To this end, 455 MRI acquisitions originating from 455 different patients that were treated at the Heidelberg University Hospital were collected as training set (HD-train). It should be noted that particularly difficult cases were preferentially included to increase the robustness of our model. Evaluation was performed on two separate test sets. The HD-test set originates from the same hospital as the training set and consists of 239 MRI acquisitions from 40 different patients (HD-test). Furthermore, to test the generalization of the algorithm, a large scale dataset comprising 2034 MRI scans from 532 patients at 34 different European institutions was collected (EORTC-test). These scans were initially acquired in the context of the EORTC-26101 study [131, 125]. The inclusion of MRI scans from multiple institutions is particularly important to test the robustness of our algorithm because MRI scanners can produce vastly differently looking images depending on the vendor, field strength

and acquisition parameters. Both test sets include multiple acquisitions per patient, each acquired at different time points, to enable evaluation also with respect to clinical parameters, such as the progression-free survival.

Each MRI scan used in the context of this study consists of four MRI modalities: T1w, T1c (T1w with Gadolinium contrast agent), T2 and FLAIR. In the context of deep learning, these modalities are simply treated as different color channels (just like red, green and blue in natural images).

Prior to annotation, all images were transformed to the standard MNI orientation and coregistered to the T1w image. Skull stripping [132] was performed on the T1w image, corrected manually if necessary, and the resulting mask was transferred to the remaining modalities. The resulting images show the brain region on a black background, as can also be seen in Figure 3.2.

Annotation of the enhancing tumor region as well as the edema in the HD-train and HD-test set was done semi-automatically with the ITK-SNAP [133] software as described in [21, 22]. Corrections of the form of fully manual delineations were performed whenever required. The EORTC-test set was annotated post-hoc by correcting the segmentation masks produced by an early iteration of our model. All annotations were done by radiologists with multiple years of experience. Please refer to [10] for a more thorough dataset and annotation procedure description.

3.1.2.3. Method

Preprocessing

Medical images consist of a voxel grid storing the localized intensity information of the modalities as well as geometry information that describe where and how the image is located in space: orientation, position and scale. Convolutional neural networks operate on voxel grids and cannot interpret the geometric information. To ensure compatibility, positional and rotation information is homogenized by orienting all images into MNI space (see above). Scaling (i.e. how much physical space each voxel occupies in all three dimensions) is typically heterogeneous, even within the same dataset, and must be addressed by resizing all training cases so that they have the same voxel spacing. Inspired by the BraTS dataset we select $1 \times 1 \times 1$ mm as target spacing. We resample image data with third order spline interpolation and segmentation maps with linear interpolation. Note that segmentations are first transferred into a one-hot encoding, then resampled and finally converted back to segmentation maps via argmax operation.

Unlike CT images which are quantitative and always have similar voxel intensities for the same structures, MRI image intensities are qualitative and can occupy arbitrary

value ranges. It is thus important to normalize the images to facilitate the learning and robustness of algorithms [72]. Finding good normalization techniques for MRI has received a lot of attention in the past [134]. With the emergence of deep learning-based algorithms, however, the exact normalization method has become less important as long as the intensity values are in approximately the same value range. We follow this trend and adopt the normalization technique that was also used by the BraTS 2016-winning contribution [92]: each modality is normalized separately by subtracting the mean and dividing by the standard deviation. This preprocessing technique (as well as the mean and standard deviation computation) is only applied to the brain region, leaving the outside voxels at 0. We do not apply bias field correction algorithms [135] because we found that they may negatively impact segmentation performance of large edema.

Network architecture

Our network architecture is inspired by the 3D U-Net [44] and its derivatives [25, 136]. Just like the U-Net, we follow the encoder-decoder pattern with skip connections. An overview of our network architecture is provided in Figure 3.3.

Dense encoder, lightweight decoder. The encoder aggregates the contextual information required for identifying the different classes. The decoder successively upscales this information back to the original image resolution by recombining the coarse upsampled contextual information from below with higher resolved feature maps originating from the skip connections (also see Section 2.2.3). Intuitively, the encoder therefore requires higher computational complexity than the decoder. Following this consideration, we use more convolutional layers in the encoder. To improve the gradient flow through the network, we make use of residual connections [9, 84]. Each residual block consists of two convolutional convolutional layers with kernel size $3 \times 3 \times 3$, each of which is preceded by instance normalization [137] and a leaky ReLU [138] nonlinearity. Note that our choice of the less popular instance normalization over the more commonly used batch normalization [139] is intentional: due to the small batch size the network is trained with (see below), the batch statistics used by batch normalization are unreliable and cause a degradation in performance. The encoder starts with an initial convolution that maps the four input modalities to 21 feature maps. The number of convolutional kernels (and thus the number of feature representations) is doubled with each downsampling operation. We avoid representational bottlenecks [8] by implementing downsampling via strided convolutions, allowing us to do the downsampling and increase in feature maps in one operation. Our encoder encompasses four downsampling steps, resulting in $21 * 2^4 = 336$ feature maps in the bottleneck. In the decoder, feature maps are upsampled with trilinear upsampling prior to concatenation with the features originating from the skip connections. After concatenation, a

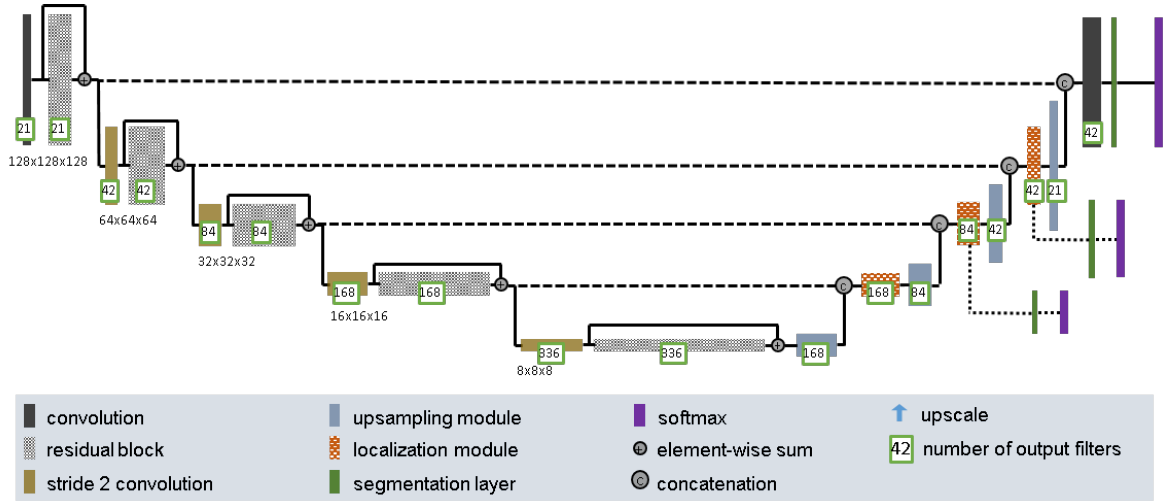

Figure 3.3.: **Network architecture brain tumor segmentation.** We use residual connections in the encoder to enable a better gradient flow and facilitate the learning of representations. The decoder is held as lightweight as possible to reduce the GPU memory footprint. Auxiliary segmentation heads are added to lower resolution stages of the decoder to encourage the training of the lower layer in the U-shape of the network. Our architecture is designed to process patches of size $128 \times 128 \times 128$ with a batch size of 2 and 21 initial feature maps. The number of feature maps is doubled with every downsampling and halved prior to upsampling. Figure reproduced from [10].

localization module is used to reduce the number of feature maps and thus the memory footprint of the network during training. It consists of two convolution-instance normalization-leaky ReLU sequences where the first convolution has a $3 \times 3 \times 3$ kernel size and the second convolution uses $1 \times 1 \times 1$ kernels. Both convolutions halve the number of feature maps (reducing the number of features to a quarter of what it was after concatenation).

Large input size The receptive field of the network, along with an appropriate input patch size, determines how much of the image the network can 'see' at once, and thus directly impacts the amount of contextual information that can be incorporated in the network's decision making. We select $128 \times 128 \times 128$ as the input patch size for the network, which covers almost an entire brain at the selected target spacing. The receptive field of the network depends on the number of downsampling operations as well as the convolutional kernel sizes and is sufficiently large in the proposed network architecture.

Deep supervision Gradient-based training of the network may push it towards finding the most simple decision rules it can represent. In the case of a U-shaped architecture such as the one here, this can cause the lower parts of the network to not be utilized effectively, ultimately encouraging the learning of decision rules solely based on the local information encoded in the upper layers. We use auxiliary loss layers at various resolutions of the decoder to push the network towards aggregating more contextual information. Auxiliary loss layers are implemented as separate low resolution segmentation outputs. During training, losses for these layers are computed with downsampled versions of the reference segmentation.

Training procedure

The network is trained for a total of 450 epochs, where one epoch is defined as 200 training iterations with a batch size of 2. Patches for constructing the minibatches are sampled randomly (with respect to the cases they are drawn from as well as their localization within the cases). We use the Adam [140] optimizer with an initial learning rate of 10^{-4}. After each epoch, the learning rate is decayed by multiplying it with 0.99.

We use a soft Dice loss [25, 24] for optimizing the network. The Dice loss inherently handles class imbalance, which is particularly important in brain tumor segmentation where the fraction of enhancing tumor voxels can be several orders of magnitude lower than that of the edema and background classes. We use the following definition of the

Dice loss:

$$\mathcal{L}_{\text{dc}} = -\frac{2}{|K|} \sum_{k \in K} \frac{\sum_{i_k} u_{i_k} v_{i_k}}{\sum_{i_k} u_{i_k} + \sum_{i_k} v_{i_k}}, \tag{3.1}$$

where u and v denote a one hot encoding of the ground truth and the softmax output of the network, respectively. $k \in K$ denotes the class identifier ($K = 0, 1, 2$ for this dataset: background, edema, enhancing tumor). $i_k \in \mathcal{N}^3$ denotes all voxels belonging to the class activation map and softmax output of the network.

The Dice loss is applied to all three output layers of the network during training. The losses are accumulated via summation, with lower weights being associated with losses originating from lower resolutions:

$$\mathcal{L}_{total} = 0.25 l_{\frac{1}{4}} + 0.5 l_{\frac{1}{2}} + 1 l_{\frac{1}{1}} \tag{3.2}$$

$l_{\frac{1}{x}}$ denotes the loss computed at $\frac{1}{x}$ of the original image resolution. The loss is computed independently for every sample in the batch and the mean loss value is used for gradient computation.

Data augmentation is a well-proven technique to improve the robustness of CNNs [141]. Overfitting is reduced by applying random transformations to the training cases during training. We use a variety of data augmentation techniques for training our brain tumor segmentation network: mirroring along all spatial axes, scaling, rotation, elastic deformation and gamma augmentation. All transformations are applied on the fly during training with randomly drawn parametrization, ensuring a large variability in the augmented training cases. See [10] for a detailed description of the data augmentation techniques used.

Inference

For inference, all test set images are preprocessed with the same pipeline as the training images. The proposed network architecture is fully convolutional, meaning that all operations in it can process arbitrary image sizes. Although $128 \times 128 \times 128$ sized patches were used for training the network, other image sizes can be processed in inference. We make use of this property of the network to predict entire scans at once, alleviating the need to stitch patches together. The resulting predictions are then resampled to their original image spacing.

Ensembling

We developed our model by running five-fold cross-validation on the HD-train dataset and optimizing its architecture as well as hyperparameters. The final training was done in a five-fold cross-validation as well, resulting in five models each trained on a slightly different training dataset. We used these five models as an ensemble to predict the two test sets. Ensembling was done by averaging the softmax outputs of the networks prior to generating the segmentation map via argmax.

Volumetry and tumor progression analysis

Once the segmentation map is available, computation of the volume of the different compartments of the tumor is straightforward. The spacing of the image gives the volume occupied by each voxel, and the number of voxels belonging to enhancing tumor and edema can be retrieved directly from the segmentation maps. Besides a change in tumor size, RANO also monitors the appearance of new lesions [17]. We detect new lesions by registering all longitudinal images in the HD-test and EORTC-test set to the first MRI scan of the respective patient. Connected component analysis on the segmentation maps from the different time steps then reveals the appearance of new lesions.

Clinical Integration

The resulting model is integrated into the clinical infrastructure. In order to not interfere with the existing pipelines, a separate Picture Archiving and Communication System (PACS) was set up. Newly acquired MRI scans are sent to both the standard clinical PACS as well as the separate PACS that runs the tumor segmentation ensuring independent and vendor-neutral operation. Our server runs the XNAT platform (`www.xnat.org`). The image processing pipeline is dockerized (`www.docker.com`) and is triggered whenever a new image arrives. The resulting segmentations and tumor volumes are reported back to the clinical PACS where they can be used alongside the raw images. For more information on the clinical integration, see details in [10].

3.1.2.4. Results

Our proposed brain tumor segmentation method shows excellent agreement with the radiologist-generated reference segmentations. Figure 3.4 shows multiple examples from the EORTC-test set highlighting both the accuracy of the model as well as the diversity of tumor appearances encountered in clinical practice. Our model is robust with respect to resection cavities and cysts, can handle multiple lesions and also works reliably when the contrast of the T1c image is low.

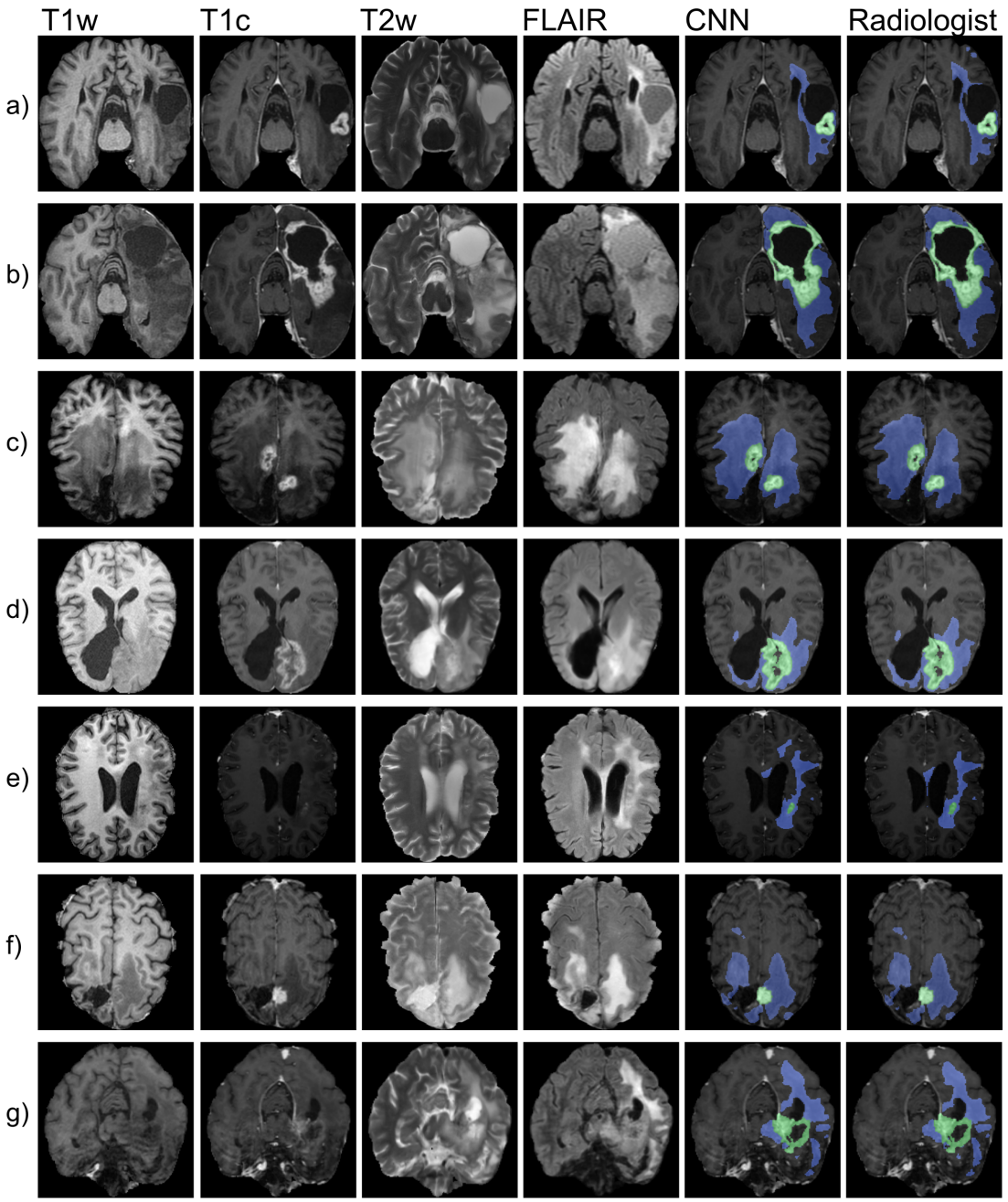

Figure 3.4.: **Qualitative results on the EORTC-test set.** Each row represents one MRI scan with the first four columns showing the four MRI modalities and the last two columns showing overlays of our predictions and the radiologist-generated reference segmentations with the T1c modality. Each row highlights challenging characteristics of the EORTC-test set: a) large cyst right next to the enhancing tumor b) tumor spanning an entire hemisphere with a large necrotic core inside the enhancing tumor region c) multiple small enhancing tumor lesions and deformation of the midline d) resection cavity e) small, barely visible enhancing tumor lesion f) small resection cavity g) low contrast of enhancing tumor, diffuse borders of the tumor and imaging artifacts.

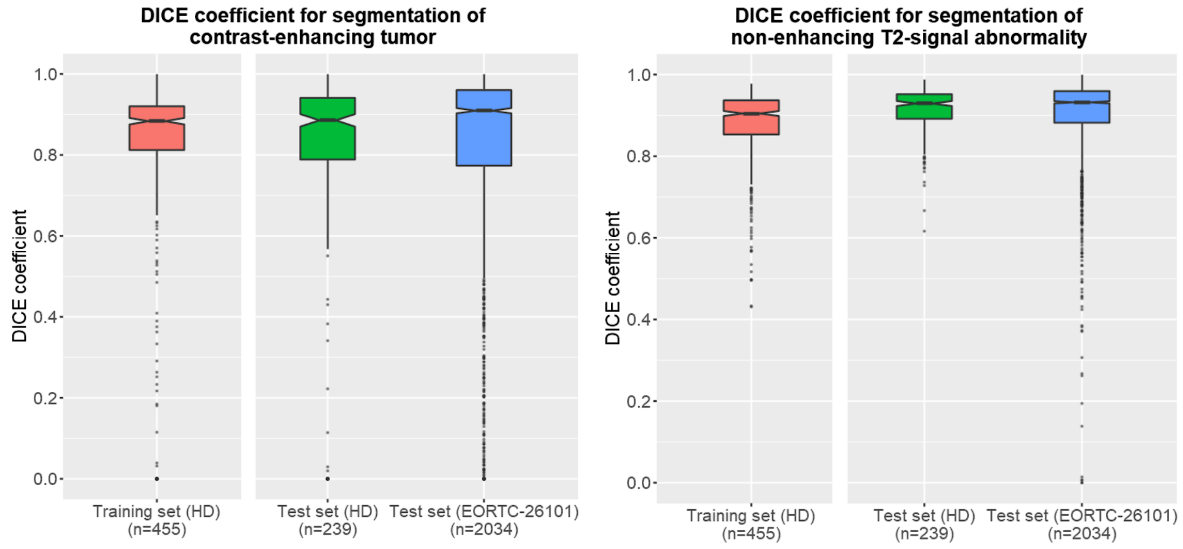

Figure 3.5.: **Quantitative results of our model on the HD-train, HD-test and EORTC-test set.** The results for the HD-train set originate from a five-fold cross-validation. The predictions for HD-test and EORTC-test were generated by ensembling the five models from the cross-validation. The midline of the boxes indicates the median value and the shaded area represents the inter quartile range. Outliers are plotted as dots. Left: enhancing tumor, right edema. Figure reproduced from [10].

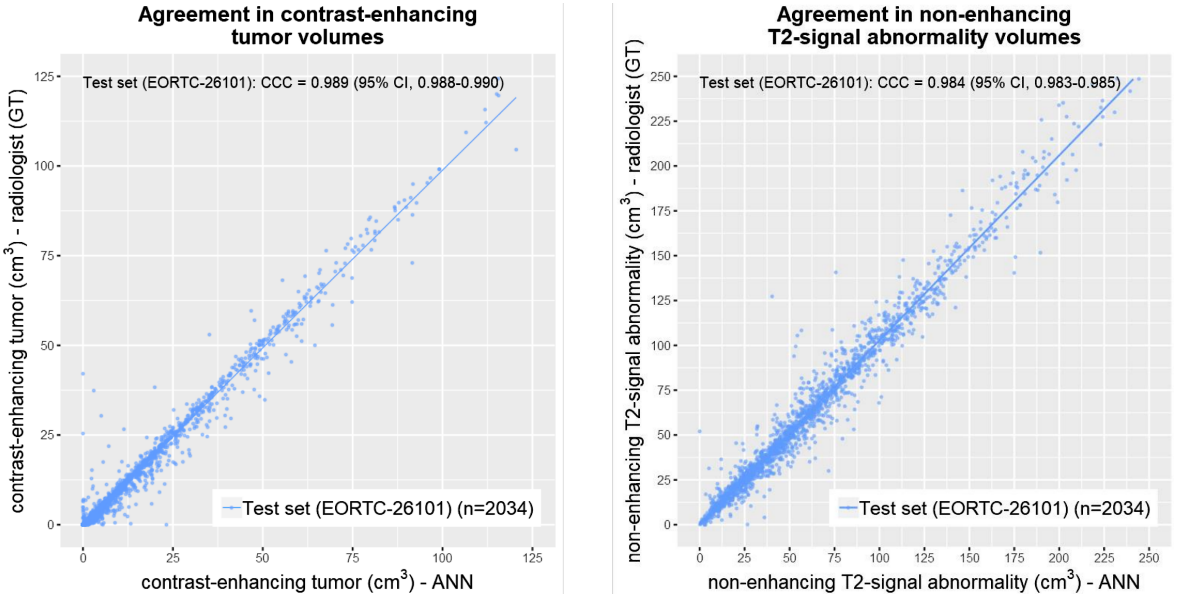

Figure 3.6.: **Volume agreement between the segmentations generated by our CNN and the reference annotation.** We observe very high agreement between the two sets of segmentations with concordance correlation coefficients of 0.989 for enhancing tumor and 0.984 for edema, underlining the value of our segmentations for the assessment of tumor therapy response. Figure reproduced from [10].

Quantitative results are provided in Figure 3.5. Our model achieves median Dice scores of 0.883 for enhancing tumor and 0.905 for edema on the HD-train set cross-validation. Our ensemble achieves median Dice scores of 0.885 and 0.906 for contrast enhancing tumor and 0.929 and 0.932 for edema on the HD-test and EORTC-test set, respectively.

The high overlap between our generated segmentations and the corresponding reference translates to an excellent volume agreement on the EORTC test set, as shown in Figure 3.6. With the focus of this thesis being first and foremost the segmentation algorithm, we refer the interested reader to our publication [10] for detailed results regarding tumor volumetry and progression analysis.

Our pretrained segmentation model is publicly available. It can be downloaded here: `https://github.com/NeuroAI-HD/HD-GLIO`.

BraTS 2017 participation

We also tested our model in a standardized and competitive environment by participating in the BraTS 2017 challenge. For this purpose we retrained our network using only the data provided by the challenge. We again make use of five-fold cross-validation and use the resulting models as ensemble.

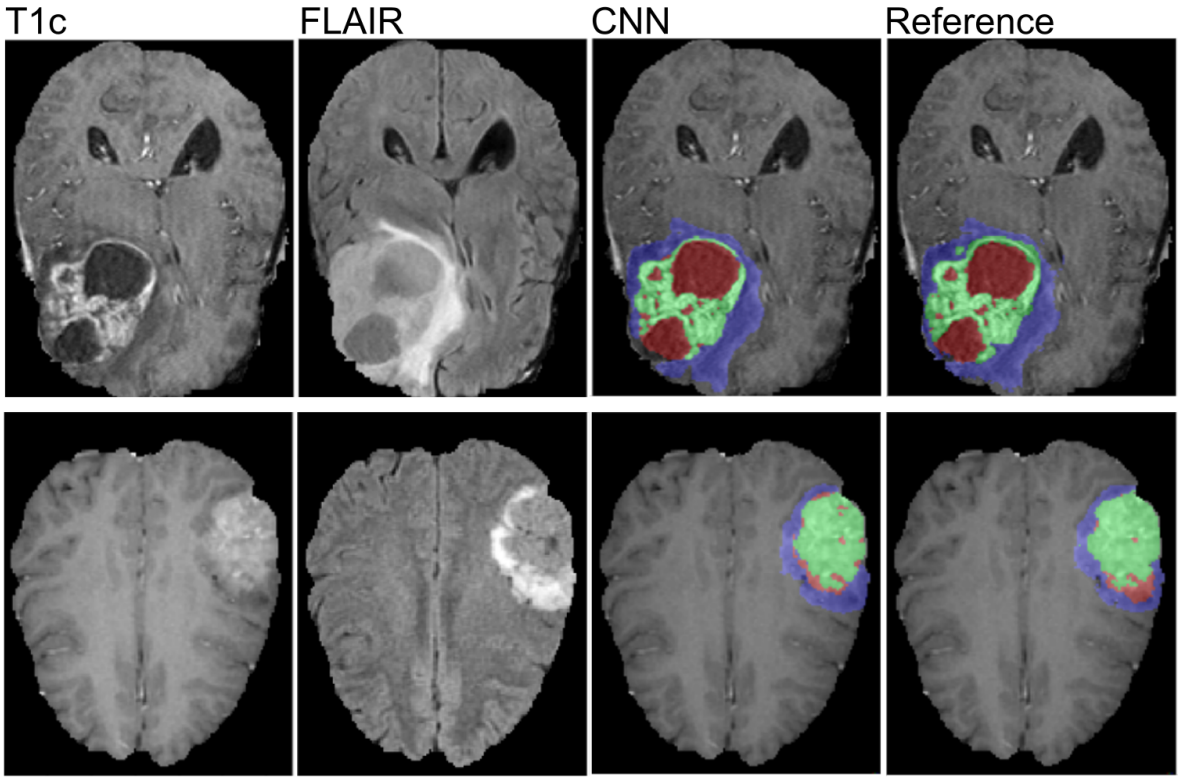

Figure 3.7.: **Qualitative segmentation results on the BraTS 2017 challenge.** Each row shows a separate example from our five-fold cross-validation on the training set of the challenge. While the first row shows excellent agreement with the reference segmentation, some disparity in the *necrosis and non-enhancing tumor* class is observed in the second example. We attribute this discrepancy to an unclear and sometimes inconsistent definition of this class within the training set. Figure reproduced from [130].

Figure 3.7 shows results from a five-fold cross-validation of our model on the BraTS training datasets. Note that BraTS has an additional label named *necrosis and non-enhancing tumor*, the definition of which is not always consistent (see Section 3.1.3.4) causing discrepancies of our prediction with the reference annotation in some training cases. BraTS evaluates predictions not on the single labels but on overlapping regions that are constructed from the labels: the whole tumor (union of edema, necrosis and non-enhancing and enhancing tumor), tumor core (necrosis, non-enhancing and enhancing tumor) and the enhancing tumor. On the training set cross-validation, our model obtains mean Dice scores of 0.895, 0.828 and 0.707 for whole tumor, tumor core and enhancing tumor, respectively. On the test set (with ground truth annotations being only available to the challenge organizers) we obtained mean scores of 0.858, 0.775 and 0.647, respectively. Among the 47 participating teams, our algorithm secured the third place [45].

3.1.2.5. Discussion

In this section we presented a model for automated tumor segmentation in MRI images. This model was trained on a large cohort of diverse tumor appearances. Evaluation both on an in-house test set as well as a large-scale multi-institutional cohort underlines the robustness of the model and emphasizes its usefulness in a clinical setting. With the emphasis of this thesis being the development of segmentation methods, details about the clinical evaluation have been left out for brevity. We refer interested readers to our publication [10] for details regarding the clinical metrics.

Perhaps surprisingly, the Dice scores of our model on the test sets surpassed the scores obtained on the training set cross-validation. We explain this unusual behavior on the HD-test set with a) the fact that the HD-train set was enriched in particularly difficult cases whereas the HD-test set is representative of the distribution found in clinical practice and b) the HD-test set being predicted with an ensemble of five models which is expected to improve the quality of the segmentations whereas the segmentations generated in the cross-validation were generated by single models. The higher scores in the EORTC-test set are certainly in part caused by these aspects, but with the reference segmentations of this set being generated post-hoc (they were generated by a previous iteration of our model and thoroughly corrected by radiologists) we cannot exclude a bias in these segmentations towards our segmentations. Therefore, the EORTC-test set should be used to evaluate the robustness of our model first and foremost. As we have shown in the Results, our model indeed produced very robust results although the images in this set originated from 34 institutions with MRI scanner parameters (vendors, field strengths, acquisition parameters) that the model had not seen during training.

We furthermore demonstrated that our model does not only produce excellent segmentation results on an in-house dataset but also in the context of an international competition. Our participation in the BraTS 2017 challenge resulted in a third place (47 teams in total) which is a respectable result given the competitive nature of the challenge as well as the lack of dataset-specific tuning.

3.1.3. Brain Tumour Segmentation Challenge 2018

This section is based on the following publication [129]:

Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., & Maier-Hein, K. H. (2018). No new-net. In International MICCAI Brainlesion Workshop (pp. 234-244). Springer, Cham.

3.1.3.1. Introduction

In the previous section (3.1.2) we presented our method for brain tumor segmentation which was developed to enable automated tumor volumetry and therapy response assessment on a large multi-institutional cohort. While the method has excellent performance both on our in-house dataset as well as the BraTS 2017 challenge, it remains unclear which design choices contributed to this effect. In this section we are going back to the 'roots': Preliminary experiments have shown that good results can also be achieved with simpler network architectures. To investigate this further, we participate in the BraTS 2018 challenge with the constraint of only using a 3D U-Net [44] like network architecture. We maximize the performance of our method through tuning of hyperparameters as well as adapting it carefully to the requirements of the competition.

The network architecture used is often treated as the defining characteristic of a segmentation method [97, 95, 25]. This seems quite surprising, especially because reducing a method to the network architecture omits all the intricacies of developing state of the art pipelines. The following section challenges this simplistic view of method development: We build our method around the 3D U-Net, an architecture that other researchers like to use as a baseline method for demonstrating the improvements that their more elaborate architecture may offer. Instead of optimizing a new approach, we invest our resources into optimizing the baseline with the goal of demonstrating that it can still achieve state of the art performance if tuned properly.

To ensure objectively good performance we evaluate the following approach in the very competitive BraTS 2018 challenge [38, 45, 142, 143, 144]. The 2018 dataset comprises 285 training cases (identical to the 2017 training set). Each case in BraTS consists of the T1w, T1c, T2 and FLAIR modalities. Note that in contrast to the in-house dataset

used in the previous section, BraTS also provides annotation for the *non-enhancing tumor and necrosis* class (see also Figure 3.7). BraTS 2018 also provides a validation set (n=66) without reference annotations. Participants can submit their predictions of the validation set to an online platform and use the obtained metrics for model development. The final evaluation is done on the test set (n=191) which is provided to the participants only shortly before the challenge deadline. Only one submission to the test set can be made. The evaluation of BraTS is not done on the raw labels but on three (partly overlapping) regions that are constructed from these labels: the whole tumor (edema, non-enhancing tumor and necrosis, enhancing tumor), the tumor core (non-enhancing tumor and necrosis and enhancing tumor class) and the enhancing tumor. Evaluation metrics are Dice score and Hausdorff distance (95th percentile) (see also 2.4).

3.1.3.2. Method

We first briefly describe our 3D U-Net-based baseline implementation followed by the improvements used to maximize performance on the BraTS 2018 dataset.

Preprocessing

The BraTS dataset is provided in a preprocessed format: all images are resampled to a common $1 \times 1 \times 1$ mm voxel spacing. The four input modalities (T1w, T1c, T2 and FLAIR) are co-registered and brain extracted with the voxels outside the brain being set to 0. We normalize the intensity values of each modality independently by subtracting its mean and dividing by its standard deviation in the brain region. The voxels outside the brain region remain 0.

Network architecture

Our network architecture, depicted in Figure 3.8, is based on the U-Net [43, 44]. It follows the successful encoder-decoder pattern with skip connections and output stride 1 (meaning that the segmentations are generated at the same size as the image and do not need to be upscaled, see Section 2.2.2). It does not use any of the recent architectural advancements and instead relies on two standard convolution-instance normalization-leaky ReLU sequences per resolution in both the encoder and decoder. The network processes $128 \times 128 \times 128$ sized patches during training with a batch size of 2. The encoder has four downsampling operations, resulting in a feature map shape of $8 \times 8 \times 8$ in the bottleneck. Due to the simpler design relative to the network used in the previous section (see 3.1.2.3) we can fit a larger number of initial feature maps in the highest resolution (30 as opposed to 21). As is convention, the number of feature maps is doubled with every downsampling operation resulting in 480 feature maps in

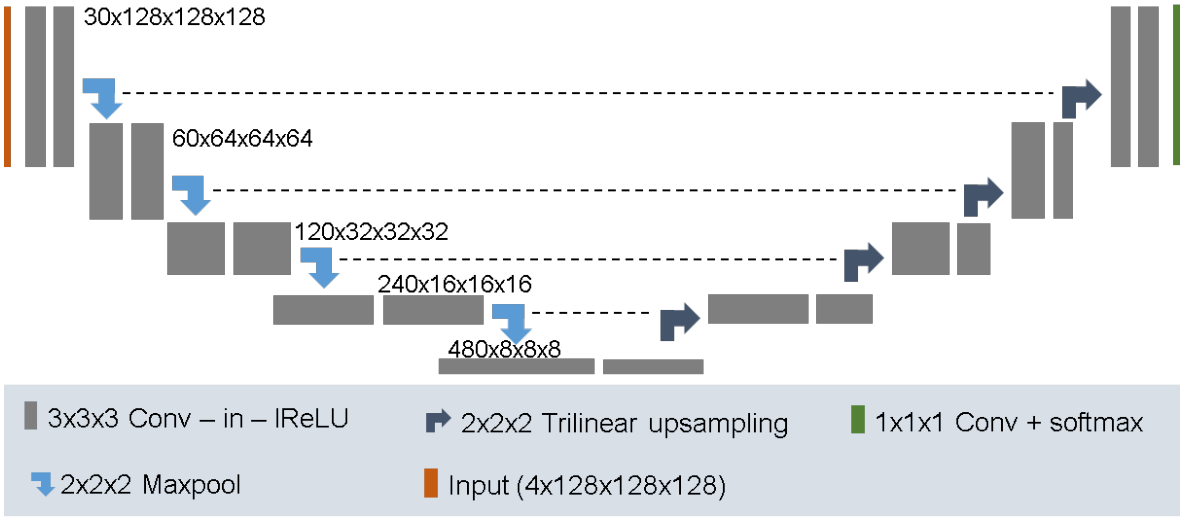

Figure 3.8.: **Network Architecture for BraTS 2018.** The network architecture used for the BraTS 2018 challenge was explicitly chosen to be standard U-Net-like. It processes patches of size $128 \times 128 \times 128$ with 30 initial feature maps.

the bottleneck. The decoder mirrors the number of features of the encoder. We justify replacing the more commonly used batch normalization [139] by instance normalization [137] with the small batch size used during training (see below). A small batch size has unreliable batch statistics and can cause issues with batch normalization [145]. We furthermore replaced the more commonly used ReLU with leaky ReLUs [138].

Training procedure

The network is trained for a maximum of 500 epochs with each epoch being defined as 250 iterations. Batches are constructed from random samples and patches are cropped randomly without oversampling rare classes. Training is done with the Adam optimizer [140] with an initial learning rate of 10^{-4}. The learning rate is reduced by multiplication with 0.2 whenever the exponential moving average of the validation loss ($l_{updated} = \alpha l_{old} + (1 - \alpha) l_{new}$; $\alpha = 0.95$) has not improved in the last 30 epochs. Training was terminated early if the validation loss did not improve within the last 60 epochs. Just like in the previous section we use the soft Dice loss for training (see Equation 3.1) to deal with the class imbalance in the dataset. During training a variety of data augmentations are applied on the fly: random rotations, scaling, elastic deformation, gamma augmentation and mirroring along all spatial axes.

Inference

We use the fully convolutional nature of our architecture to predict entire images at once. Mirroring along all axes is applied as test time data augmentation for a slight

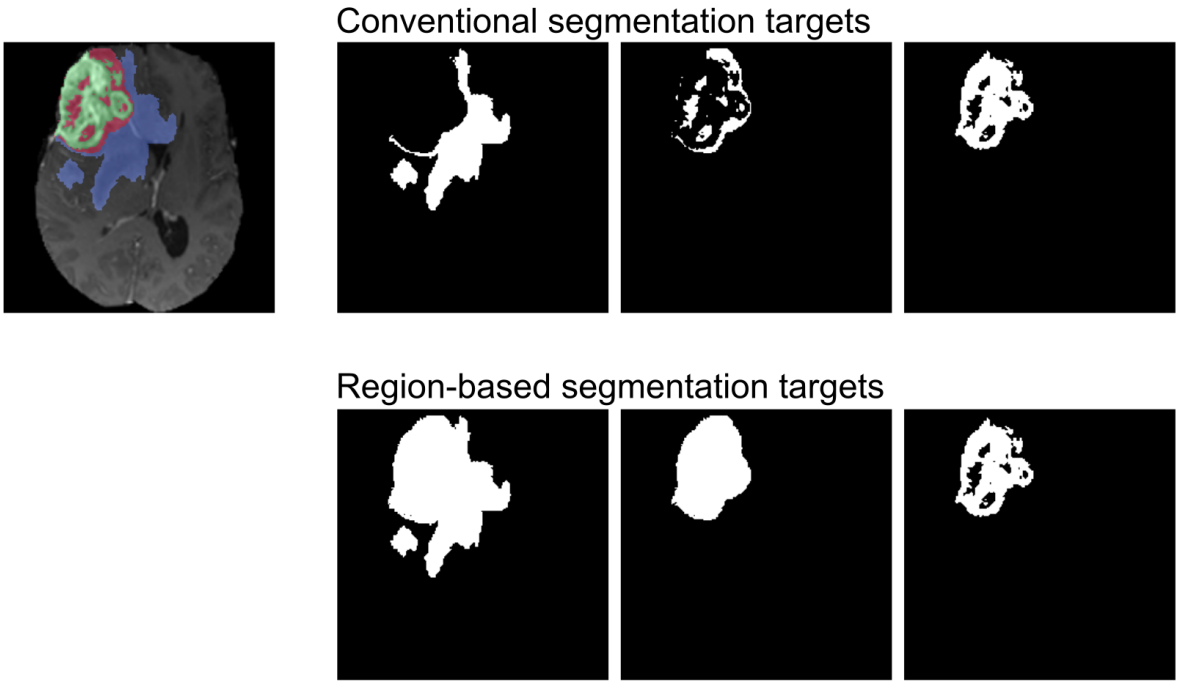

Figure 3.9.: **Region-based segmentation targets.** Left: overlay of reference annotation with the T1c image. Upper row: Conventional, mutually exclusive, segmentation targets. Bottom row: Overlapping region-based segmentations. The BraTS dataset evaluates the quality of predictions on three partially overlapping regions. To optimize for this evaluation scheme as well as putting less emphasis on the inconsistently annotated *non-enhancing tumor and necrosis* class, we optimize the regions directly.

increase in accuracy. Whenever used, ensembling is implemented by averaging the softmax probabilities (or sigmoid outputs, see below).

Improvements over the baseline implementation

Region-based segmentation targets The BraTS evaluation is made on three partially overlapping regions instead of the raw labels. Optimizing the regions directly as opposed to indirectly could improve the segmentation accuracy because the network would learn to optimize the target metrics directly.

Figure 3.9 depicts the idea behind region-based training. To the left is an overlay of the three labels that are provided by the training data. The upper row shows the conventional, mutually exclusive segmentation targets. During training, the logits of the segmentation network are passed through a softmax nonlinearity and the loss function (here a multiclass Dice loss) is applied to transformed logits. This forces the network to have a final embedding where the three labels are pushed into the corners of

a hypercube, enabling linear separability and pushing the classes as far away as possible from each other. We believe that this approach could cause issues because it does not respect the hierarchical nature of the data (the enhancing tumor is a part of the tumor core which in turn is a part of the whole tumor). Region-based training is depicted in the bottom row. Making the network output the hierarchical regions directly optimizes the target metric and also puts less emphasis on the *non-enhancing tumor and necrosis* class which is ill-defined and annotated inconsistently. In region-based training, this class is no longer optimized directly and only plays an indirect role as part of the whole tumor and tumor core. Our region-based training is inspired by [146] who designed a cascade of CNNs to segment the three regions one after the other. To simplify the segmentation method we incorporate the region optimization into a single network. Overlapping segmentation targets are accommodated by replacing the final softmax layer with a pixelwise sigmoid which makes our outputs no more mutually exclusive. We construct the reference regions from the ground truth annotations. Optimization is done as before with a soft Dice loss (the soft Dice loss does not require mutually exclusive labels because it treats the ground truth as one-hot or multi-hot and works independently for each output).

Co-training with external data Although 285 cases is plenty in the context of medical image segmentation, additional training data could improve the results even further. When using additional data, these can be either used for pretraining or along with the available training data. The additional training data would need to follow the same annotation convention as the original data. Due to the annotation procedure of the BraTS 2018 dataset [45], neither the in-house dataset from the previous section or previous BraTS datasets [38] can be used for this purpose naively. Still, some similarities in the annotation procedure exist and could be used. To prevent contamination of our BraTS predictions with different annotation procedures we use external data with an approach similar to 'M Heads' [147]. We add an additional segmentation layer at the end of our network that has a separate set of weights to generate the segmentations from the previous feature representation. The samples originating from BraTS are directed towards the BraTS-specific segmentation output whereas the samples originating from the additional data source are directed towards their segmentation head (we only use one external data source at once). All remaining network weights and representations are shared. During training, we use one sample taken from the BraTS dataset and one from the external dataset per minibatch. We experiment with the Task01_BrainTumour from the Medical Segmentation Decathlon [29] as well as the in-house dataset from the previous section as external training data.

Postprocessing Metric aggregation in BraTS overemphasizes zero false positives in the enhancing tumor class. The following equation recaps the definition of the Dice

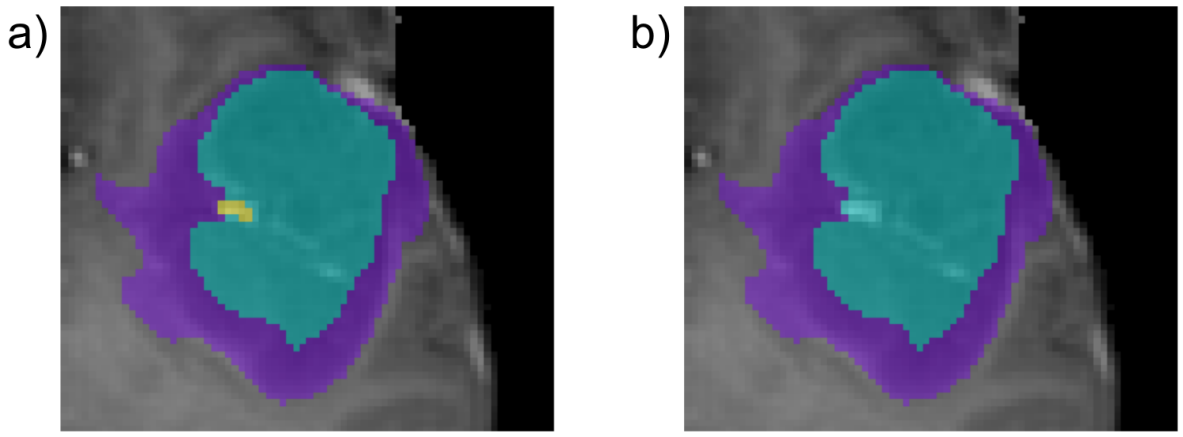

Figure 3.10.: **Postprocessing by removing small enhancing tumor regions.** BraTS awards predictions with no false positives when there is no enhancing tumor in the reference segmentation with a Dice score of 1. We introduce a simple postprocessing technique that exploits this by discarding the enhancing tumor prediction entirely when only small enhancing tumor regions are predicted. Left: overlay of our prediction before postprocessing. The enhancing tumor is shown in yellow, the edema in purple and the non-enhancing tumor and necrosis class in turquoise. This example demonstrates one of the most common failure cases in brain tumor segmentation: blood vessels still containing contrast agent are erroneously segmented as enhancing tumor. In the BraTS evaluation, this case would have gotten a Dice score of 0 for this class. Right: Postprocessing removed the false positive prediction, awarding a Dice score of 1 for the non-enhancing tumor class.

score:

$$DSC = \frac{2A \cap B}{|A| + |B|} \tag{3.3}$$

with $A \cap B$ being the intersection between two segmentations A and B and $|A|$ and $|B|$ being the number of elements in A and B, respectively. If both reference and prediction do not contain the enhancing tumor class, all numbers in the above equation are zero, causing the Dice score to be undefined. Instead of excluding these cases from the metric aggregation, BraTS handles this case by assigning a Dice score of 1, rewarding the participants for their lack of false positives (note that a single false positive voxel will cause the Dice score to be 0!). If one considers that small lesions usually have lower scores because small mistakes have a large effect on the Dice, it may be beneficial to trade an increase in false negatives for the large reward that is a Dice score of 1 by postprocessing the predictions such that small predicted enhancing tumor regions are dropped.

Table 3.1.: **Results on BraTS 2018 training data (285 cases).** All results were obtained by running a five fold cross-validation. Metrics were computed by the BraTS online evaluation platform. reg: region-based training. cotr: cotraining with additional training data. post: postprocessing by removing small enhancing tumor regions. DC&CE: using sum of Dice and cross-entropy as loss function (as opposed to Dice only).

	Dice (higher is better)			HD95 (lower is better)		
	enh.	whole	core	enh.	whole	core
Isensee et al. (2017) [130]	70.69	89.51	82.76	6.24	6.04	6.95
baseline	73.43	89.76	82.17	4.88	5.86	7.11
baseline + reg	73.81	90.02	82.87	5.01	6.26	6.48
baseline + reg + cotr (dec)	75.94	91.33	85.28	4.29	4.82	5.05
baseline + reg + cotr (dec) + post	**78.68**	91.33	85.28	3.49	**4.82**	**5.05**
baseline + reg + cotr (dec) + post + DC&CE	78.62	**91.75**	**85.69**	**2.84**	4.88	5.11
baseline + reg + cotr (inst) + post + DC&CE	76.32	90.35	84.36	3.74	5.64	5.98
baseline + reg + post + DC&CE	76.78	90.30	83.55	3.66	5.36	6.03

We experiment with increasing the score obtained for the enhancing tumor class by removing the enhancing tumor class entirely from an image if the total volume of the predicted enhancing tumor is less than some threshold. Enhancing tumor is removed by replacing it with necrosis (as shown in Figure 3.10). The associated threshold is tuned on the training set by testing several thresholds and picking the one that resulted in the highest aggregated Dice score.

Loss function improvement The Dice loss, while optimizing the target metric directly and inherently handling class imbalance [25, 24, 148] can be difficult to optimize for neural networks [148]. We experiment with complementing it with a pixel-wise cross-entropy loss. The loss terms are simply merged with summation. No weighting is applied to them.

3.1.3.3. Results

We train and evaluate our baseline model as well as its derivatives on a five-fold cross-validation on the training cases as well as the official BraTS validation set. The validation set is predicted using the five models from the training set cross-validation as an ensemble. Note that the final evaluation of the challenge is done on the test set, for which only one submission could be made.

Table 3.1 gives an overview of the performance of our baseline model as well as its derivatives on the training set cross-validation. Due to the BraTS 2018 training cases being equivalent to the ones used in 2017, we can also directly compare the results to our previous model (which was introduced in section 3.1.2). Although our previous model had a much more elaborate network architecture, the simple U-Net baseline used in this

Table 3.2.: **Results on BraTS2018 validation data (66 cases).** Results were obtained by using the five models from the training set cross-validation as an ensemble. Metrics were computed by the BraTS online evaluation platform.

	Dice			HD95		
	enh.	whole	core	enh.	whole	core
baseline	79.59	90.80	84.32	3.12	4.79	8.16
baseline + reg + cotr (dec) + post + DC&CE (*)	80.46	91.21	85.77	2.52	4.38	6.73
baseline + reg + cotr (inst) + post + DC&CE (**)	80.95	91.15	86.6	2.44	5.02	6.73
baseline + reg + post + DC&CE	80.66	90.92	85.22	2.74	5.83	7.20
ensemble of (*) and (**)	80.87	91.26	86.34	2.41	4.27	6.52

project outperformed it by a quite substantial margin on the enhancing tumor class. Region-based training (reg) yields a small improvement over the baseline model in all evaluation regions. Cotraining with the data from the Medical Segmentation Decathlon (cotr (dec)) yields a substantial improvement on the training set. Postprocessing in the form of removing enhancing tumor for cases where the total predicted enhancing tumor volume was below $750mm^3$ improved the scores of the enhancing tumor class even further. Complementing the Dice loss with a cross-entropy loss improved the scores on the whole tumor and tumor core. Interestingly, using our institutional data (cotr (inst)) performed worse than the cotr (dec).

The results on the validation set are summarized in Table 3.2. Surprisingly, cotr (inst) performed substantially better then cotr (dec) on this set. This is the opposite of what was observed for the training set cross-validation. With only one test set submission available, this discrepancy is problematic. Due to the larger size of the training set (n=285 vs n=66) one could lean towards favoring the cotr (dec) model. However, as we have used the training set extensively for model development and hyperparameter tuning, the validation set may give a better indication of the test set performance. We finally opted for ensembling the two models. Even though the ensemble did not yield noticeably better performance than the cotr (inst) model on the validation set we hypothesized that it would be more robust and therefore selected it for the test set prediction.

Figure 3.11 shows a qualitative example taken from the validation set. All tumor classes seem to have been delineated accurately. Notably, the blood vessels located to the left of the enhancing tumor region (the bright structures in the T1c image) have not been falsely segmented. The thin rim of *non-enhancing tumor and necrosis* voxels around the enhancing tumor region is an artifact of region-based training and does not adversely affect performance.

Our algorithm obtained the second place out of 61 competing teams in the BraTS 2018 challenge. The winning method by Andrij Myronenko [99] outperformed our method

Figure 3.11.: **Qualitative results on the BraTS 2018 validation set on a particularly difficult case.** All classes have been delineated accurately. Edema is shown in purple, enhancing tumor in yellow and *non-enhancing and necrosis* in turquoise. Figure reproduced from [129].

Table 3.3.: **Test set results.** We show the scores achieved by NVDLMED [99], the winner of BraTS2018, and our method ('MIC-DKFZ'), which achieved the second place. Table reproduced from [129].

		Dice			Hausd. dist.		
		enh.	whole	core	enh.	whole	core
NVDLMED	Mean	76.64	88.39	81.54	3.77	5.90	4.81
	StdDev	25.57	11.83	24.99	8.61	10.01	7.52
	Median	84.41	92.06	91.67	1.73	3.16	2.45
MIC-DKFZ	Mean	77.88	87.81	80.62	2.90	6.03	5.08
	StdDev	23.93	12.89	25.02	3.85	9.98	8.09
	Median	84.94	91.79	90.72	1.73	3.16	2.83

in whole tumor as well as tumor core while we achieved higher Dice scores in the enhancing tumor class. [45] provides a detailed overview of the test set results.

3.1.3.4. Discussion

In this section we investigated how far we can push the segmentation performance of a simple 3D U-Net-derived network architecture. By optimizing the hyperparameters as well as introducing optimizations for the BraTS 2018 dataset to the training and post-processing pipeline we were able to obtain the second place in the highly competitive BraTS 2018 competition. In a research field where overwhelming research effort is put into the advancement of network architectures, this is a remarkable result because it strictly opposed the direction research in the field is taking.

We should note that the postprocessing technique in the context of this method was developed solely to improve the challenge metric. It should never be used in a clinical environment where the accurate detection and delineation of small enhancing tumor regions can have a substantial impact on the prognosis and treatment of patients. We would like to see metric aggregation in BraTS to be changed so that they better reflect clinical requirements. This can be achieved by simply excluding the cases with no enhancing tumor in both prediction and ground truth from metric aggregation (but retaining them if false positive voxels are present).

Transitioning from BraTS 2016 to later challenges, the organizers have merged the non-enhancing and necrosis classes into a single *non-enhancing tumor and necrosis* class. The non-enhancing class in particular was not well defined and had little evidence in the images, causing the annotations to be inconsistent and hard to reproduce. Starting with BraTS 2017, the organizers have identified this problem: "In order to address the aforementioned issue, in BraTS 2017 the NET label ("Label 3") has been eliminated and combined with NCR ("Label 1")." [45]. However, by combining these two classes together, the underlying issue remains and inconsistent annotations in the training set can cause issues preventing good model optimization. Region-based training seems to be an effective approach for dealing with this issue by putting less emphasis on optimizing this label.

The drop in Dice scores in the test set relative to the performance on the training and validation set is a recurring sight in the competition. It is unclear as to what causes it, but several observations point towards a general problem associated with the test set itself. We believe it highly unlikely that this drop is related to any of the design choices in our method. We base this hypothesis on two pillars. First, all participating teams experience this drop in performance making it highly unlikely to be caused by overfitting in our method. Second, in the context of this thesis, we will present a

large number of highly competitive segmentation methods. The vast majority of these methods are evaluated on holdout test sets for which the reference annotations are not accessible. While all of these methods were developed using the same principle (tuning on five-fold cross-validation on the training cases), none of these methods experience a similar drop in test set performance.

3.1.4. Discussion

This chapter summarized our efforts in advancing brain tumor segmentation methods. In Section 3.1.2, we have investigated tumor volumetry and therapy response assessment on a large cohort. Although the model was trained on an in-house dataset, it proved to be robust when applied to a large cohort of cases originating from 34 different institutions across Europe. We also applied this algorithm to the BraTS 2017 challenge where we achieved the third place out of 47 competing teams. In Section 3.1.3 we have questioned our elaborate network architecture and designed a simple 3D U-Net-derived segmentation method. Instead of focusing on architectural advancements, this method was optimized solely through hyperparameter optimizations as well as training and postprocessing pipeline improvements for good performance on the BraTS 2018 dataset. Interestingly, even the simple baseline model could outperform our previous network architecture. By incorporating several improvements we could successively improve the segmentation accuracy of the model and ultimately achieved the second rank out of 61 competing teams.

3.2. Heart Segmentation

This section is based on the following publication [27]:

Isensee, F.*, Jaeger, P. F.*, Full, P. M., Wolf, I., Engelhardt, S., & Maier-Hein, K. H. (2018). Automatic Cardiac Disease Assessment on cine-MRI via Time-Series Segmentation and Domain Specific Features. Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges (pp. 120–129). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-75541-0_13

(*: shared first authorship)

Note that this section only describes my contribution to this publication, which was the development, implementation and evaluation of the segmentation method. The disease classification method, which was contributed by my co-author Paul F. Jaeger, is explicitly omitted.

3.2.1. Motivation

Structural changes in the heart can cause heart failure if they remain untreated. Reliable identification of structural alterations is essential not only for diagnosis and treatment stratification but also for managing patients at risk [149, 150, 151]. Clinical processes are hereby tightly wound around digital imaging techniques, such as Cardiac Magnetic Resonance Imaging (MRI). On the one hand, cardiac cine MRI (CMRI) offers high contrast in soft tissue allowing the measurement of the relevant compartments in the heart. On the other hand it allows to capture a time series of the moving heart in 3D, enabling the analysis of the dynamics of the heartbeat. Clinical analysis of CMRI images starts with manual or semi-automated segmentation of the end systolic (ES) and end diastolic (ED) images [152]. Based on the segmentation masks as well as the underlying images, quantitative parameters characterizing the heart are extracted[4]. Ejection fractions (EF) of the ventricles describe what percentage of their end diastolic volume (i.e. when the heart is relaxed) is ejected when the heart contracts. The stroke volume complements this value by providing the absolute volume difference between ED and ES. Quantification of the myocardial wall thickness between left and right ventricle allows insights into possible prior infarction. These well-established metrics are regularly used in rule-based clinical decision processes and guidelines [153].

Although cardiac segmentation has been an active research area in recent years [154, 155], the accuracy and robustness of the resulting models does not yet meet clinical standards. Therefore, the required segmentations are still created either manually or with the help of semi-automatic tools. This process is not only time consuming, but also introduces substantial inter-rater and even intra-rater variability [152] resulting in insufficiently reliable outcomes.

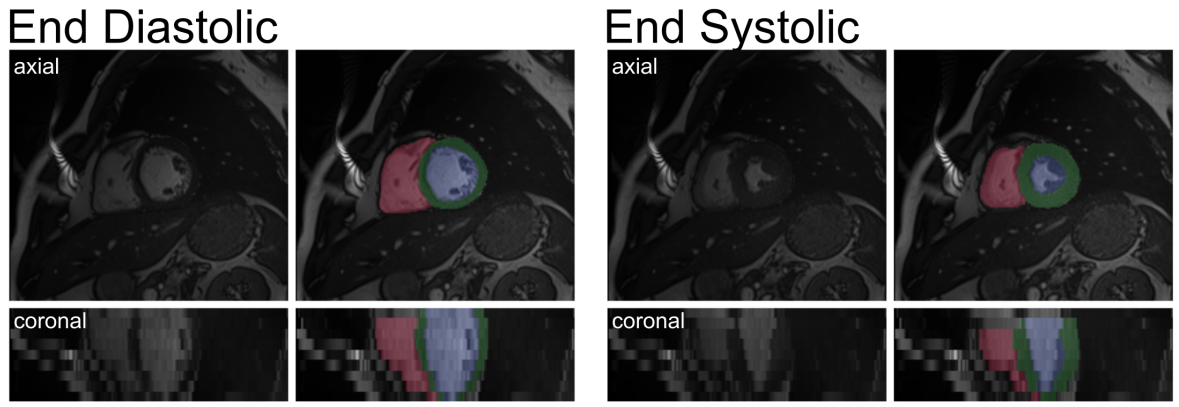

Figure 3.12.: **ACDC training data.** This example shows the end diastolic and end systolic time steps of *patient_066* from the ACDC challenge [4]. The axial view in the top row demonstrates the high in-plane resolution. The coronal view in the bottom row highlighting the low out of plane resolution. The left image shows the raw data and the right image shows an overlay of the segmentation maps on the raw data. The right ventricle is shown in red, the myocardium of the left ventricle in green and the left ventricular cavity in blue.

3.2.2. Introduction

The Automated Cardiac Diagnosis Challenge (ACDC) [4] was created to encourage and enable the development of automated cardiac segmentation methods. It provides a large training cohort of 100 CMRI time series. For each time series, the left ventricular cavity (LVC) the right ventricle (RV) and the myocardium of the left ventricle (LV¡) are segmented in the ES and ED time step, resulting in a total of 200 annotated images. One unique feature of the ACDC dataset is its inclusion of patients with pathologies. The training set consists of patients with abnormal right ventricles, infarction, hypertrophic cardiomyopathy, dilated cardiomyopathy and a control group of healthy patients (20 cases each). The dataset provides 50 test cases with unknown pathologies and segmentations for independent method evaluation.

Figure 3.12 shows an example image taken from the ACDC training dataset. One of the major challenges associated with this dataset is the disparity between in and out-of-plane resolution. As is typical for CMR images of this type, the slice-wise 2D acquisition results in a high in-plane resolution but suffers from inter-slice distances of up to 1 cm. Higher resolutions in the out of plane axis are certainly desirable but difficult to achieve with patients suffering from coronary diseases because they require longer image acquisition times during which the patients are required to hold their breath. Furthermore difficulties associated with this dataset are slice misalignments due to patient movement between slice acquisitions, the presence of trabeculae and papillary muscles inside the LVC, partial volume effects as well as banding and motion

artifacts [4].

3.2.3. Method

Our approach for CMR segmentation uses an ensemble of standard 2D and 3D U-Net architectures [43, 44] that were specifically adapted to the ACDC dataset.

3.2.3.1. Preprocessing

All images were normalized individually by subtracting their mean value and dividing by their standard deviation. The ACDC dataset contains training cases with varying voxel spacings. Neural networks, which operate on voxel grids, cannot encode this information. Therefore, all images need to be resampled to the same voxel spacing prior to feeding them into the network. We selected $1.25 \times 1.25 \times 10$ mm as target spacing for the 3D U-Net. For the 2D U-Net the in-plane target spacing is set as 1.25×1.25 mm as well. Since the network operates on in-plane slices, the out-of-plane spacing is left unchanged to alleviate the need for resampling across the low resolution axis. Resampling is done with linear interpolation. Segmentation maps are converted to a one-hot encoding prior to resampling and converted back with the argmax operation.

3.2.3.2. Network Architecture

Figure 3.13 gives an overview of the segmentation network used for the ACDC challenge. Note that both the 2D and 3D network follow the same global topology but differ in their implementation. The encoder consists of five blocks, with each block being composed of $2\times$(convolution-batch normalization-leaky ReLU) [138, 139]. Downsampling is done with max pooling. The number of feature maps doubles with each downsampling operation. The decoder mirrors the structure of the encoder. Upscaling feature maps is implemented as linear upsampling. Features originating from the skip connections are concatenated to the upsampled feature maps. The decoder is augmented with deep supervision, similarly to [136]: additional segmentation outputs are generated at lower resolutions in the decoder, upsampled to match the full resolution output and aggregated by addition prior to softmax activation. All convolutions are padded to ensure that they have identical input and output shapes.

Both the 2D and 3D U-Net are configured to fill the GPU memory of a 12 GB Nvidia Titan X during training. The 2D U-Net uses an input patch size of 352×352, processing almost entire axial slices. The initial number of feature maps is set to 48, allowing a batch size of 10. The 3D U-Net processes batches of size $10 \times 224 \times 224$ with a batch size of 4. All kernel sizes of feature map generating convolutions are set to be 3×3 in the 2D U-Net and $3\times3\times3$ in the 3D U-Net. Due to the anisotropy of the batches in

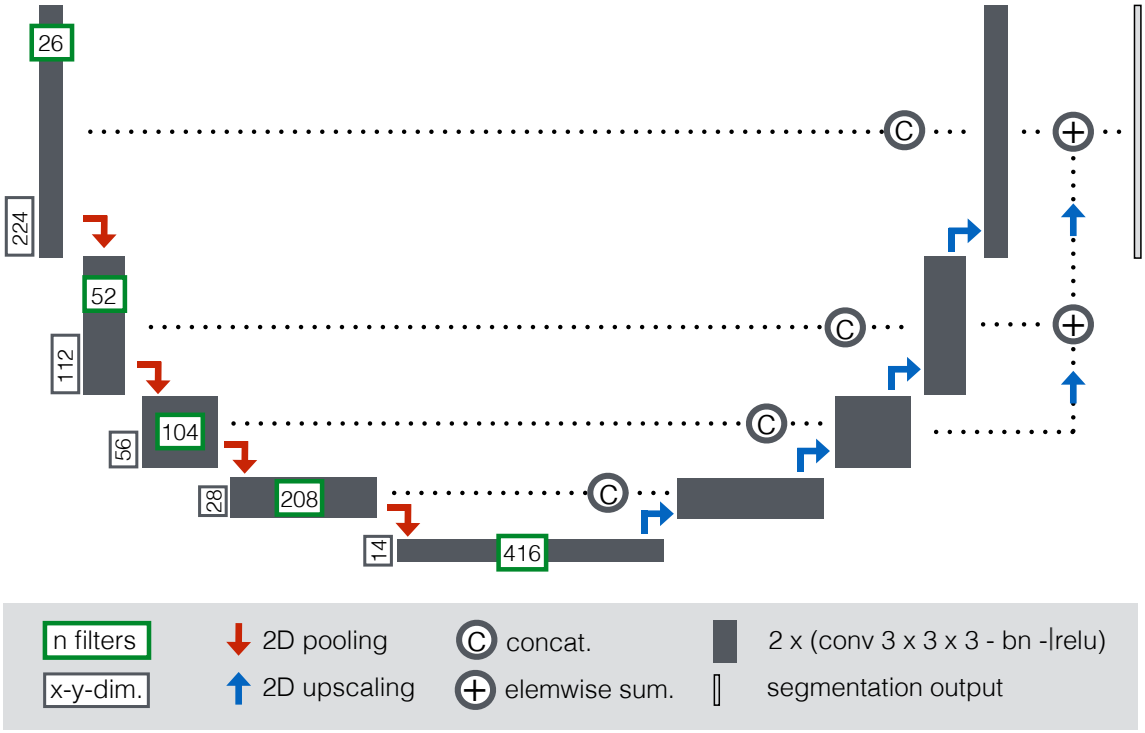

Figure 3.13.: **3D U-Net architecture for the ACDC challenge.** The input patch size of the network is $10 \times 224 \times 224$ voxels. Due to the anisotropy of the patch, pooling throughout the entire network is only done in-plane, resulting in the out-of-plane dimension to remain at 10 at all times. Information between slices are aggregated only through the $3 \times 3 \times 3$ convolutional kernels. The network starts with 26 feature maps which double with every downsampling step in the encoder. The decoder mirrors the number of feature maps in the encoder. We introduce additional auxiliary segmentation outputs at lower resolutions in the decoder which are upsampled and added to the segmentation output of the highest output resolution. The 2D network architecture is equivalent but uses 2D convolutions and starts with a patch size of 352×352 and 48 feature maps. Figure reproduced from [27].

the 3D network, downscaling (and upscaling) is only performed in-plane, leaving the size of the out of plane feature maps untouched at a size of 10 voxels. Aggregation of contextual information between slices thus occurs only through the convolutional kernels in the 3D U-Net (and not at all in the 2D U-Net).

3.2.3.3. Training procedure

Both the 2D and 3D model were trained in a five-fold cross-validation to obtain estimates of their performance on the training cases. The splits of the cross-validation were generated such that the ED and ES image of a patient were either both part of the training or the validation split (patient-level stratification).

The 3D model was trained for 300 epochs with each epoch being defined as 100 iterations with batch size 4. The batches were constructed by randomly selecting cases from the training split and randomly cropping patches out of these images. Categorical cross-entropy was used as loss function. We used the Adam optimizer [156] with an initial learning rate of $5 \cdot 10^{-4}$ which was decayed by multiplication with 0.98 after each epoch.

Just like the 3D model, the 3D model was also trained for 300 epochs with each epoch being defined as 100 iterations. Adam with the same settings was used here as well. Patches were sampled by selecting random slices from random training images and then cropping the slices randomly to the desired input. The 2D network was trained with a multiclass variant of the Dice loss [25, 24]:

$$\mathcal{L}_{\text{dc}} = -\frac{2}{|K|} \sum_{k \in K} \frac{\sum_{i_k} u_{i_k} v_{i_k}}{\sum_{i_k} u_{i_k} + \sum_{i_k} v_{i_k}}, \tag{3.4}$$

with u and v being a one hot encoding of the ground truth and the softmax output of the network, respectively. $k \in K$ denotes the class identifier ($K = 0, 1, 2, 3$ for the ACDC dataset: background and three different foreground classes LVC, MLV, RV). $i_k \in \mathcal{N}^3$ denotes all voxels belonging to the class activation map and softmax output of the network, both having a shape of $10 \times 352 \times 352$. Note how k extends into the batch dimension, effectively treating the individual samples in the minibatch as a pseudo-3D volume to increase the robustness of the loss.

A variety of data augmentation techniques are applied during training to increase the robustness of the networks to unseen data: mirroring along the in-plane axes, random rotations, gamma augmentation and elastic deformations. Due to the anisotropy of the data resampling along the out of plane axis results in resampling artifacts. To prevent

Table 3.4.: **Comparison of 2D and 3D U-Net performance on the ACDC training set.** Dice scores computed from a five-fold cross-validation. Table reproduced from [27].

	LVC	RV	LVM
2D model	0.945	0.902	0.905
3D model	0.928	0.879	0.872
ensemble	**0.945**	**0.908**	**0.905**

artifacts being introduced by data augmentation we apply all spatial transformations only in-plane. We furthermore artificially increased the number of slice misalignments for the 3D network by shifting slices with a probability of 10% by a random offset drawn from a Normal distribution $\mathcal{N}(0, 20)$. The offset is sampled independently for x and y.

3.2.3.4. Inference

We apply the same preprocessing techniques used for preparing the training cases to the test cases. Note that this entails generating two sets of preprocessed test images: one for the 3D U-Net and one for the 2D U-Net. Prediction of the images was done fully convolutionally. The softmax outputs were then sampled back to the original image resolution. The final segmentations were obtained by averaging the softmax predictions of the 2D and 3D model followed by an argmax operation. Note that for each U-Net model we use the five models obtained from cross-validation in the ensemble, resulting in a total of 10 models for the final test set predictions.

3.2.4. Results

3.2.4.1. Cross-validation results

The Dice scores obtained on the five fold cross-validation on the ACDC training dataset are presented in Table 3.4. Although the 2D model outperforms the 3D model in all three labels, ensembling the two models improves the segmentations of the RV label and supports our design decision to submit the ensemble of these models for the test set.

3.2.5. Test set results

The ACDC test set consists of 50 patients, again each with ED and ES time points for a total of 100 images. We predicted the test cases with the inference methodology described in Section 3.2.3.4.

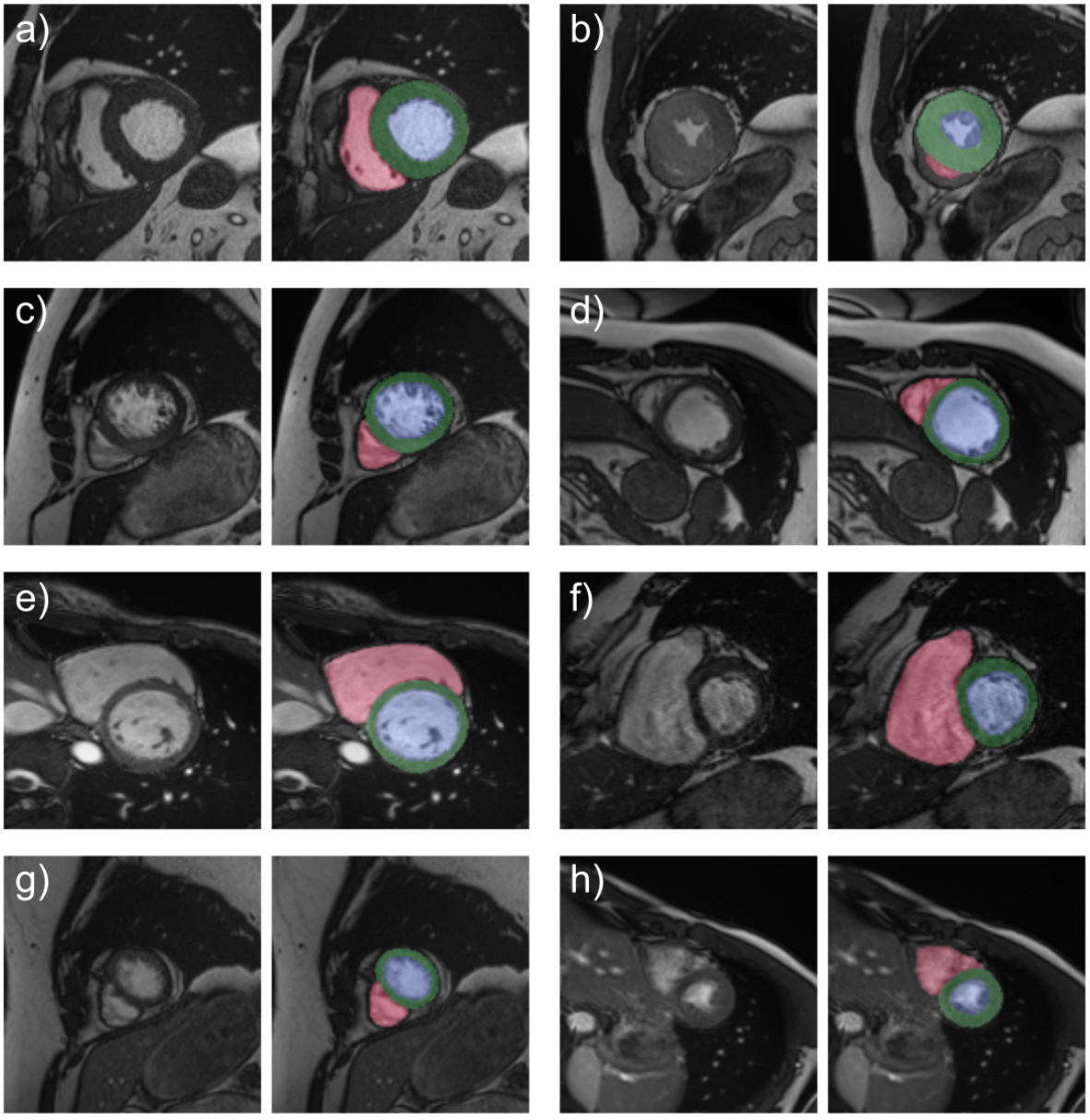

Figure 3.14.: **Test set results for ACDC.** Each plot shows a different test case with the raw image on the left and an overlay with our prediction on the right.

Table 3.5.: **Quantitative ACDC test set results.** All scores were computed and provided by the challenge organizers [4].

	LVC	RV	LVM	Average
Isensee et al. [27]	0.9495	0.9225	0.9105	0.9275
Baumgartner et al. [157]	0.937	0.9075	0.8965	0.9137
Khened et al. [94]	0.9405	0.907	0.8935	0.9137
Zotti et al. [158]	0.931	0.9115	0.89	0.9108
Jang et al. [159]	0.94	0.907	0.885	0.9107
Wolterink et al. [160]	0.9395	0.9	0.8845	0.908
Rohe et al. [161]	0.9285	0.8805	0.8815	0.8968
Jain et al. [162]	0.92	0.865	0.8895	0.8915
Tziritas-Grinias [163]	0.9065	0.803	0.7975	0.8357
Yang et al. [164]	0.8195	0.7795	N/A	N/A

Figure 3.14 shows eight representative samples of the test set along with the predictions of our model. Although no ground truth masks are available for comparison (these are only available to the challenge organizers) visual inspection reveals high segmentation accuracy. This is even true for difficult cases, for example in the upper right corner (b) where the right ventricle (red) is barely visible due to the pathological appearance of the heart (likely hypertrophic cardiomyopathy). The models also seem to be robust with respect to shadows introduced by blood flow (f) [4] and motion artifacts (h).

The quantitative test set results of all teams that participated in the 2017 challenge are presented in Table 3.5. As can be seen in the table, the proposed ensemble of 2D and 3D U-Net outperformed all competing methods by a fair margin and thus won the ACDC challenge. Note that all scores were computed and provided by the challenge organizers [4]. We added a column showing the average score for the three regions for convenience.

Note that our Dice scores for all three classes (RV, LV, LVM) were higher on the test set than on the training set cross-validation. While ensembling the five models from cross-validation certainly plays a role (due to ensembling 10 predictions per image on the test set as opposed to 2 on the training set), this result still emphasizes the robustness of our model with respect to previously unseen images.

Our segmentation method is publicly available. Source code can be downloaded here: `https://github.com/MIC-DKFZ/ACDC2017`.

3.2.6. Discussion

In this section we presented our method for cardiac MRI segmentation in the context of the ACDC challenge. Our method revolves around adapting the successful 2D and

3D U-Net architectures to deal with the specific difficulties encountered in the ACDC dataset, the most important of which being the strong anisotropy of the data (high in-plane and low out-of-plane resolution). Even though our approach is based on plain convolutions, and thus had one of the simplest feature extractors among the challenge competitors, the careful design of the method allowed it to outperform substantially more sophisticated approaches [94].

Perhaps surprisingly, the 3D U-Net performed worse than its 2D counterpart on the five fold cross-validation. We would have expected it to perform substantially better because it can transfer information between slices and thus better cope with situations in which structures cannot be recognized using a single slice (see for example Figure 3.14 b). Note that other participants also investigated the use of 3D U-Nets and made similar observations about their performance relative to 2D architectures [157]. We attribute the lower segmentation accuracy of the 3D network to two properties of the ACDC data. First, the large out-of-plane resolution causes substantial changes between successive slices that may go beyond what the small $3 \times 3 \times 3$ kernel sizes of the convolutions can process. Second, slice misalignments worsen this effect substantially by introducing adjacent slices where the structures are shifted substantially relative to each other. Both of these effects may introduce a substantial amount of noise into the convolution operations hampering the learning process of the networks and causing them to overfit more than their 2D counterparts.

Surprisingly, despite its overall worse performance, ensembling the 3D U-Net with the 2D U-Net improved the Dice score of the most difficult class, the RV, while not impacting the Dice scores of the other classes. We suspect that despite its shortcomings, the 3D network was able to learn complementary information to the 2D network that enabled it to improve upon the latter's performance in the RV class.

Finally, we should note that the resampling process selected for this method may, even though it was tuned towards anisotropic data, still be suboptimal. Some images in the training set had substantially better out of plane resolution (5mm) than the majority of images (10mm). Downsampling these images to a common out of plane spacing of 10mm and the upsampling the predictions back to 5mm can cause substantial interpolation artifacts, especially in the presence of slice misalignments. It would have been beneficial to select a higher out-of-plane resolution as target spacing resulting in less downsampling during preprocessing.

3.3. Kidney and Kidney Tumor Segmentation

This section is based on the following publication ([165]):

Isensee, F., & Maier-Hein, K. H. (2019). An attempt at beating the 3D U-Net. arXiv preprint arXiv:1908.02182.

3.3.1. Motivation

Between 1983 and 2002 the incidence of kidney tumors has risen from 7.1 to 10.8 cases per 100,000 US citizens [166]. The frequency of small incidental findings in particular has increased, which is largely attributed to the widespread availability of imaging techniques [166]. Surgical removal of the tumors is curative and is considered the standard treatment approach [167], but comes with associated adverse effects to the patients health.

Treatment of renal tumors has advanced substantially in the last 60 years, going from radical nephrectomy (i.e. complete removal of the affected kidney) [168] to kidney-preserving partial nephrectomy [169]. Recent evidence suggests that a substantial number of tumors may even be indolent, meaning that they are unlikely to ever become dangerous for the patient [170, 171]. For these cases, active surveillance constitutes the best treatment option, is considered safe [171] and provides the highest patient well-being. To identify the best type of treatment and thus outcome for each individual patient, proper stratification techniques are required. The distinction of malignant renal cell carcinoma from benign kidney tumors is, however, considered difficult on radiological images [172]. Nonetheless, several scoring systems for quantification of tumor aggressiveness have recently been proposed [18, 19], but so far lack widespread adoption in clinical practice. This is partly due to high manual labelling effort [173], substantial inter-observer variability [174] as well as insufficient accuracy of the predictions [175, 176, 177].

Semantic segmentations of the kidneys and tumors have the potential to drastically improve the accuracy and inter-rater variability of these scoring systems. However, simply due to the time required to do the delineation manually, this has so far not been considered to be an option for clinical practice. For this reason, fully automated segmentation methods could have tremendous impact on clinical decision making by taking away the manual effort while at the same time providing a way to generate reproducible, high quality segmentation masks. Furthermore, automatic segmentations can be deployed on a large scale at minimal cost, ultimately enabling the discovery of more precise and robust scoring systems for treatment stratification. Although substantial advances in automatic segmentation methods on CT images, most prominently in liver and liver tumor segmentation [26, 35, 29] as well as multi-organ segmentation

[36] have been made in recent years, kidney tumor segmentation remains an unsolved problem with only few methods being devoted to it [178, 179, 180].

3.3.2. Introduction

The Kidney and Kidney Tumor Segmentation Challenge (KiTS) [15, 28] was created to encourage the development of automated segmentation methods for kidneys and kidney tumors and to identify the best algorithm for this task. It was held in conjunction with the Conference for Medical Image Computing and Computer Assisted Interventions (MICCAI) in 2019. The challenge provides the largest fully annotated dataset for this type of problem to date with 210 training and 90 test images. The segmentations for the training cases are released to the public while the annotations of the test set are held private and used for method evaluation.

Reference annotations comprise labels for the Kidneys as well as the tumors. Figure 3.15 shows two examples from the training set. The kidneys are shown in purple and the tumors are shown in yellow. Tumors can be identified via their texture and based on the deformation they cause, causing the kidneys to bulge outward. One of the major difficulties in the dataset is the presence of cysts (see bottom row in the Figure) that can be difficult to distinguish from tumors.

The U-Net [44, 43] and its derivatives are the de facto state of the art in most medical image segmentation applications [99, 34, 27, 129, 26]. As we have discussed in Section 2.2.4, many newly proposed methods attempt at improving upon the U-Net by introducing alterations to its architecture [97, 95, 94, 93, 25]. However, results both from previous sections in this thesis (see Sections 3.2 and 3.1.3) as well as recent challenge results [30, 129] indicate that state of the art results can be still achieved with just a U-Net, questioning the necessity for complex design patterns in segmentation architectures in the medical domain. In this section we will be revisiting this question by directly comparing a standard U-Net with U-Nets that incorporate residual connections in their encoder.

3.3.3. Method

3.3.3.1. Preprocessing

The voxel spacing of the images provided by KiTS is heterogeneous and must be homogenized for processing with neural networks. Selecting the proper target spacing is crucial because it changes the size of the images as well as how much fine-grained details are discernible after resampling. A larger target spacing results in smaller images with less detail whereas a smaller spacing results in larger images with more details. When working with CNNs, the amount of contextual information that can be used by the

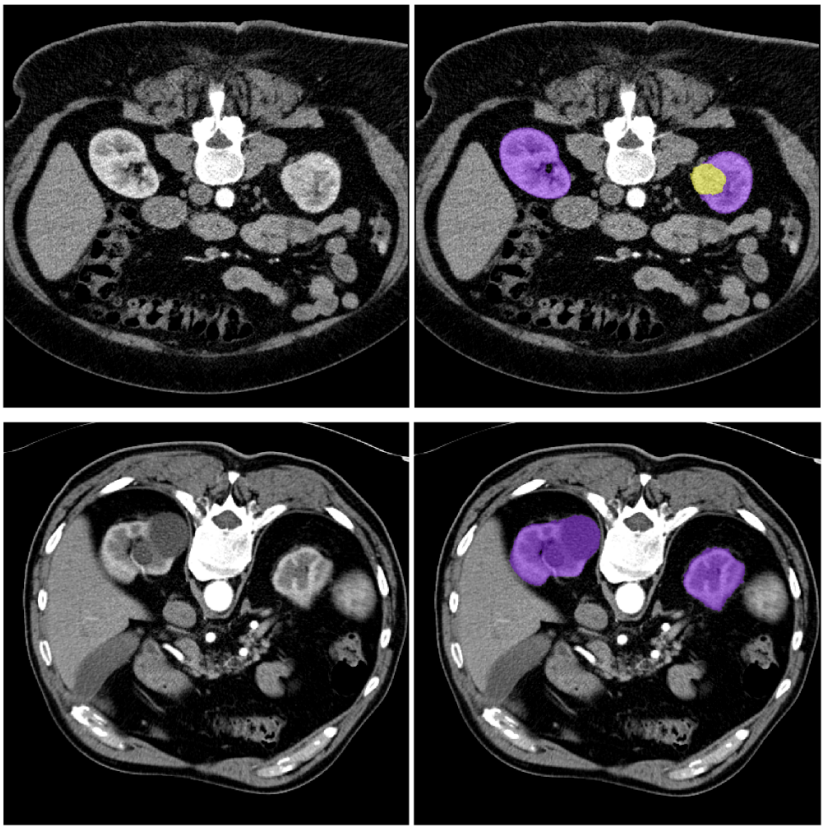

Figure 3.15.: **Kidney and Kidney tumor segmentation.** Examples are taken from the KiTS 2019 training set. Left: raw CT image, Right: overlay with the reference segmentation. Kidneys are shown in purple, tumor in yellow. Tumors can be difficult to discern from kidney tissue and may be identified either by protruding from the kidney or by their texture (upper row). Cysts can look similar to tumors (bottom row), making them one of the main sources for segmentation errors in this dataset.

network is determined by the spatial extend the input patch size can cover (provided that the network architecture is designed appropriately). Ideally, we would like to set the input patch size to be as large as (or close to) the image size to enable the network to see all the relevant details. This is, however, often impossible in practice because large patch sizes require a large amount of GPU memory during training. Based on our experience in Section 3.1.3, an input patch size of $128 \times 128 \times 128$ can reasonably be processed by common deep learning capable GPUs, such as the Nvidia Titan X (12 GB) or Nvidia RTX 2080 ti (11 GB). With the input patch size being limited by the GPU memory, the proportion of the image the network can see at once depends on the target spacing because it determines how many millimeters the patch size corresponds to. Maximum contextual information would be accessible to the network if the target spacing was chosen such that the resulting average image shape was approximately equal to the patch size. However, excessive downsampling comes at the price of less fine grained texture information being retained, possibly decreasing the accuracy with which structures that are best identified by their texture are segmented. Furthermore, downsampling the reference segmentations to the new spacing also causes a loss in fine grained details and resampling the result back to the original image resolution introduces interpolation artifacts. We therefore strike a balance between image size, texture quality and the granularity of the segmentations by selecting a target spacing of $3.22 \times 1.62 \times 1.62$ mm which results in a typical shape of the training cases of $128 \times 248 \times 248$ (median size in voxels, computed individually for each axis).

CT images are quantitative with voxel intensities representing Hounsfield Units (HU), a measure of how much radiation is absorbed by the tissue. Unlike MRI images which produce qualitative image intensities, HU values are comparable between different CT scanners. This property is used by clinicians for the analysis of CT images: organ-specific *level windows* can be set to clip undesired values and increase the perceived contrast within the structure of interest. We follow this idea in our intensity normalization scheme by clipping the intensity values of all images to the range $[-79, 304]$. We then subtract 101 and divide by 76.9 to bring the resulting intensities into a range that can easily be processed by CNNs.

3.3.3.2. Network architecture

In this section we compare the segmentation accuracies of a standard 3D U-Net with two residual variants: a 3D U-Net with residual connections [9] and a 3D U-Net with pre-activation residual connections [84] in the encoder. Since residual networks facilitate the construction of deeper network architectures through improved gradient flow and the inherent capability of learning residual functions, we increase the number of convolutional layers in the residual variants to reflect this advantage. In the context of

medical image segmentation, the GPU memory used for training is the most limiting constraint when designing segmentation networks, which is why we configure all network architectures to fully utilize the memory of a 12 GB Titan X GPU to allow for a fair comparison. It is important to note that the number of convolutional layers and parameters intentionally differ between the architectures.

Figure 3.16 gives an overview over the U-Net architectures used in this section. The upper part shows the standard 3D U-Net and the lower part the residual U-Nets. The residual U-Nets share the same topology with only the type of residual blocks being different. Both architectures process a patch size of $80 \times 160 \times 160$ voxels and have 5 downsampling operations. The first downsampling operates only in-plane to account for the anisotropy in the patch size. This results in a feature map size of $5 \times 5 \times 5$ in the bottleneck. Downsampling is implemented as strided convolution and upsampling is implemented as convolution transposed. To reduce the memory footprint of the residual U-Nets, they use only one convolution per resolution in the decoder as opposed to 2 in the standard U-Net. The standard U-Net starts with 30 feature maps and the residual U-Nets with 24.The number of feature maps is doubled with each downsampling, up to a maximum of 320. Auxiliary loss layers are added to the decoder (also see Section 3.1.2.3).

3.3.3.3. Training procedure

The training procedure is identical for all three U-Nets. We use stochastic gradient descent with nesterov momentum and a batch size of 2. Training is done for 1000 epochs with one epoch being defined as 250 iterations. Following our experience in Section 3.1.3 the sum of cross-entropy and multiclass Dice loss are used. During training we apply a variety of data augmentation techniques on the fly: scaling, rotation, brightness, gamma, contrast and Gaussian Noise.

3.3.3.4. Dataset Modifications

During the training phase of the challenge the organizers confirmed the reference segmentation of two cases (IDs 15 and 37) to be faulty [1]. We therefore replaced the segmentations of these cases with the ones generated from initial runs of our standard U-Net. During model development, we furthermore noticed consistent disagreement between our predictions and four additional cases (23, 68, 125 and 133) which led us to exclude them for the final experiments reducing the number of training cases to 206.

[1] `https://github.com/neheller/kits19/issues/21`

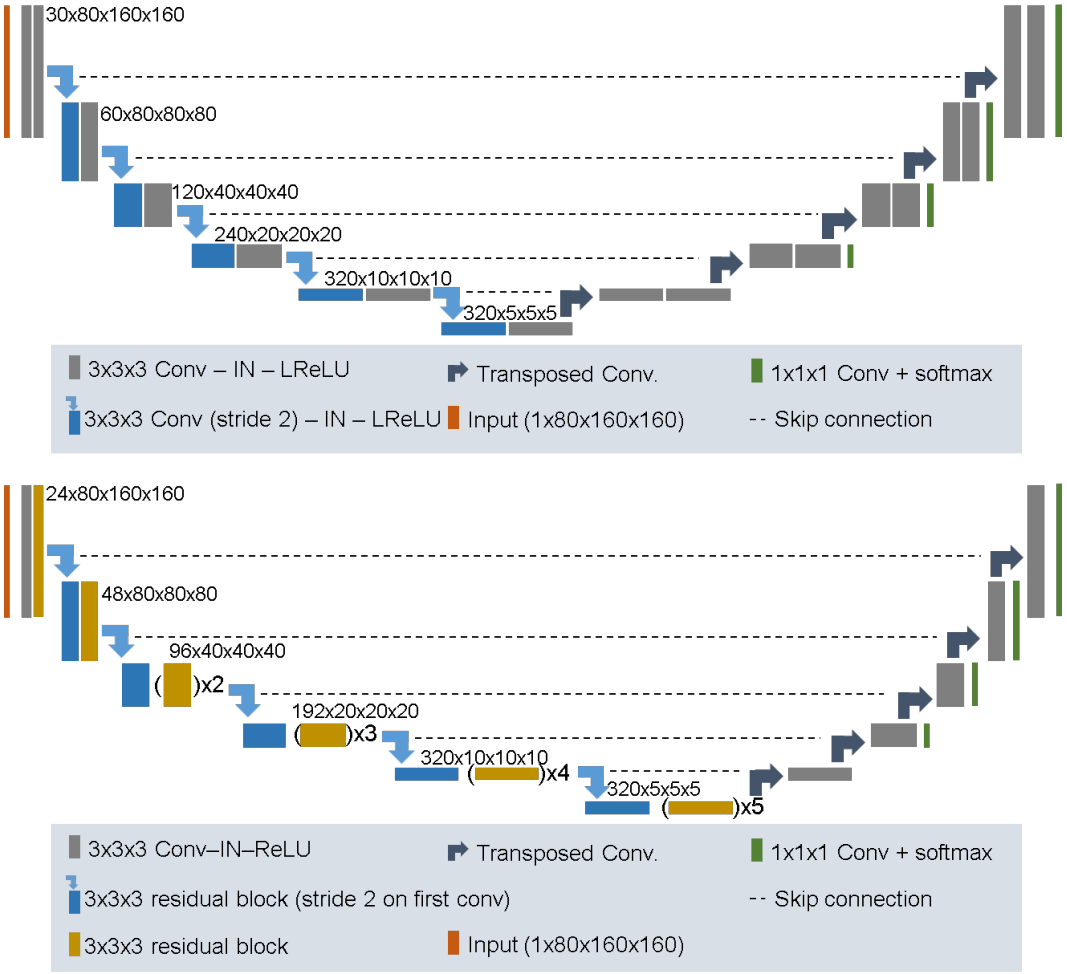

Figure 3.16.: **3D U-Net and its residual counterparts used for the KiTS 2019 challenge.** We investigate whether including residual connections can improve the segmentation accuracy of the U-Net. The reference U-Net implementation uses two convolutions per stage in both the encoder and decoder (upper row). The residual and pre-activation residual U-Nets (bottom row, they only differ by the type of residual block being used) have a deeper encoder to utilize the benefits of using residual connections, namely the improved gradient flow and the possibility of creating deeper network architectures. The decoder of the residual U-Nets only uses one convolution per stage to offset the increased memory consumption of the encoder. They also start with only 24 feature maps instead of 30 to fit the same memory budget as the reference U-Net during training. Figure reproduced from [165].

Table 3.6.: **Five-fold cross-validation on the KiTS 2019 training data.** The residual 3D U-Net achieved the highest composite Dice score and is therefore selected for test set prediction.

Network architecture	Kidney Dice	Tumor Dice	Composite Dice
3D U-Net	97.34	85.04	91.19
Residual 3D U-Net	97.36	**85.73**	**91.54**
Preact. Res. 3D U-Net	97.37	85.13	91.25
ensemble of all U-Nets	**97.43**	85.58	91.50

3.3.3.5. Inference

Due to the disparity in patch and image size ($80 \times 160 \times 160$ vs typical image size of $128 \times 248 \times 248$) we do not use fully convolutional inference (as was done in Sections 3.1.2, 3.1.3 and 3.2) and instead make use of a sliding window approach. We set the window size to be the same as the patch size that was used during training and scan the images. Since the segmentation accuracy decreases towards the borders we use 50% overlap between adjacent patches and weigh the voxels in the center of each prediction higher than the ones close to the border. This ensures that during the aggregation of the single predictions pixels that have a higher confidence influence the final prediction the most. Whenever ensembling is used we compute the softmax probabilities for each model separately and average them across the ensemble members.

3.3.4. Results

We train all three U-Net models in a five fold cross-validation on the 206 training cases to obtain a reliable estimate of their segmentation performance. Note that the KiTS challenge ranks methods by their *composite Dice*, which is defined as the average of the kidney and tumor dice scores. In case of a tie the tumor dice is used as tie breaker [15]. Table 3.6 shows the Dice scores of our experiments with the three U-Net variants. All networks show excellent agreement with the reference annotations, yielding composite Dice scores of 91.19 for the standard U-Net, 91.54 for the residual U-Net and 91.25 for the pre-activation residual U-Net. We attempted increasing the segmentation scores by ensembling the three models. Interestingly the ensemble could not improve upon the single model score of the residual U-Net.

Based on how close the scores for the U-Nets are it is difficult to declare a winner, especially because there is always a certain amount of noise in the results. Nonetheless, one model had to be selected for test set prediction. For this we selected the residual U-Net based on its marginally higher composite Dice. We used the five models from the cross-validation as an ensemble for predicting the 90 cases of the test set.

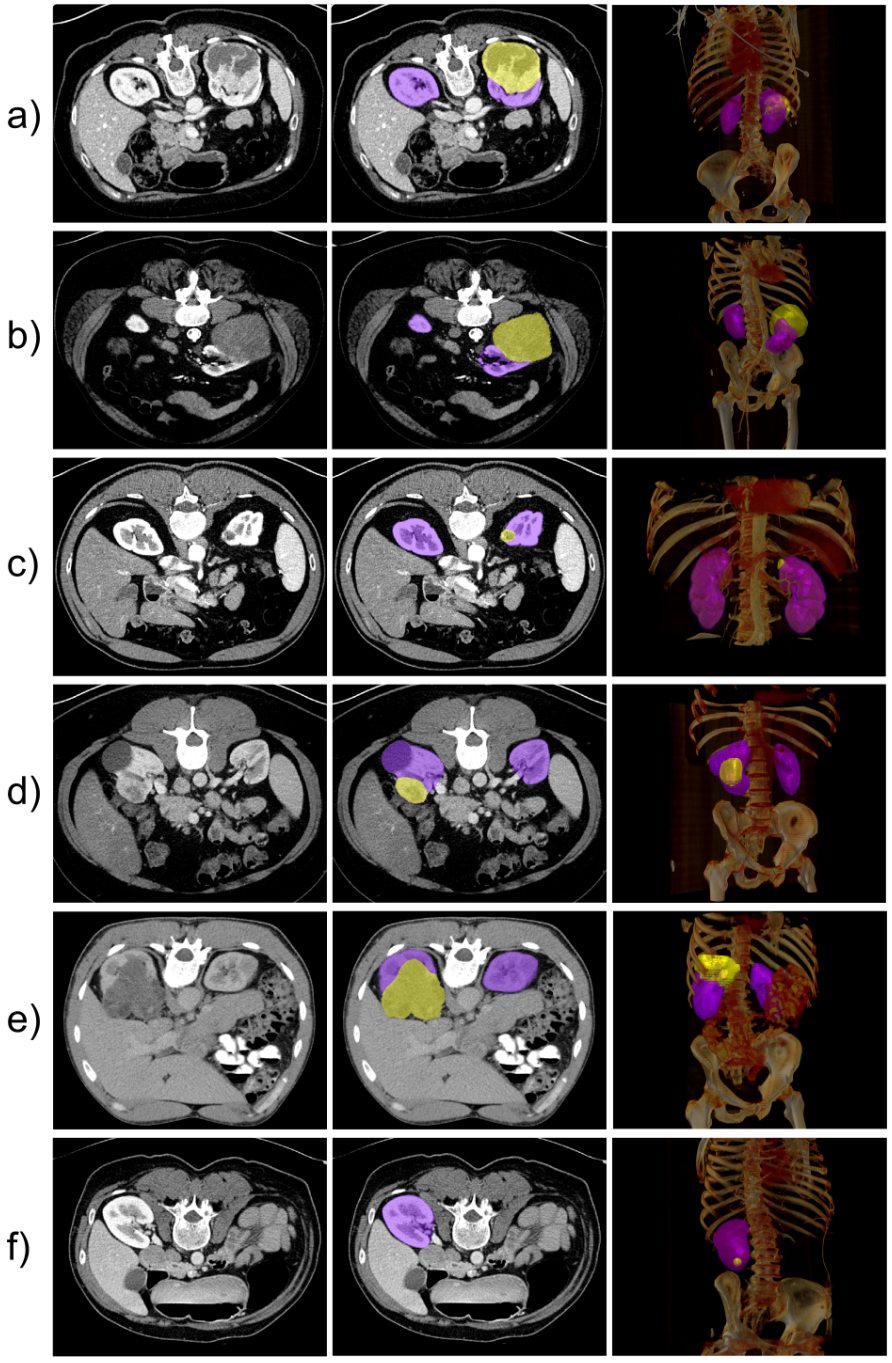

Figure 3.17.: **Qualitative segmentation results on the KiTS 2019 test set.** Left: raw CT image; Middle: Our predicted segmentation; Right: 3D Rendering. Kidneys are shown in purple and tumors in yellow. Each row highlights a particular difficulty of the dataset. a) Large heterogeneous tumor and kidney displacement. b) homogeneous tumor and strong kidney displacement. c) small tumor that is difficult to discern from the kidney. d) cyst and tumor. e) large tumor that protrudes from the kidney. d) one kidney missing. Renderings are done with the MITK workbench [181].

Table 3.7.: **Results on the KiTS 2019 test set (90 cases).** Our segmentations were generated by using the five residual 3D U-Nets obtained from the training set cross-validation as an ensemble. Only the top 10 entries to the challenge are shown for brevity. The full leaderboard can be accessed at `http://results.kits-challenge.org/miccai2019/`

Rank	Team	Composite Dice	Kidney Dice	Tumor Dice
1	Isensee F and Maier-Hein K. H.	91.23	97.37	85.09
2	Xiaoshuai H. et al.	90.64	96.74	84.54
3	Guangrui M. et al.	90.25	97.29	83.21
4	Zhang Y. et al.	90.24	97.42	83.06
5	Ma J.	89.94	97.34	82.54
6	Liu S.	89.87	97.42	82.31
7	Wszhao_fdu	89.61	97.41	81.81
8	Li Y.	89.39	97.17	81.61
9	Myronenko A. and Hatamizadeh A.	89.23	97.42	81.03
10	Chen B.	89.20	97.01	81.40

Figure 3.17 shows qualitative results of our method on the KiTS 2019 test set. With the reference predictions being held private by the challenge organizers we can only show our predictions and not compare them to the ground truth. Visual inspection reveals a high segmentation quality. The figure highlights particularly difficult cases with cysts, kidney displacement, missing kidneys as well as particularly large and small tumors.

For the evaluation of the challenge the predicted segmentations were submitted to the challenge organizers. In the final ranking, our method outperformed all 105 competitors and took the first place with a composite Dice score of 91.23 and a margin of 0.59 to the second place.

3.3.5. Discussion

In this section we have developed three 3D U-Net variants for kidney and kidney tumor segmentation. The three variants, namely a standard U-Net, a U-Net with residual blocks in the encoder as well as a U-Net with pre-activation residual blocks in the encoder. We trained all architectures in a five-fold cross-validation on the 206 training cases and selected the best performing method, the U-Net with residual connections for the test set submission. Our method obtained the first place in the KiTS 2019 challenge, which is a notable achievement considering that KiTS was the most competitive challenge at this venue with 106 participating teams.

Even though the residual U-Net ended up being selected for the test set predictions, its performance on the training set cross-validation relative to the standard U-Net was

surprisingly similar to the standard U-Net implementation. The difference between the models is only 0.35 Dice scores, which is in the same range we would expect the noise on this dataset to be. This difference is also less than the margin to the second place in the challenge, suggesting that the standard U-Net implementation could also have achieved the first place had it been used for the test set prediction.

Training the same model several times will always result in a certain amount of noise in the scores. The experiments done in this section therefore do not allow a definitive conclusion about which U-Net model really performs the best on the KiTS dataset. The residual U-Net variants both performed marginally better then the standard U-Net, but without rerunning the experiments several times and doing a statistical analysis of the distribution of scores no definitive conclusions can be drawn from this study. While this may appear disappointing at the first glance, it definitely is a win for the standard U-Net whose performance was much higher than the literature would suggest. This is particularly surprising considering that the residual U-Nets have more parameters and more convolutional layers, which should have allowed them to learn more expressive features [9, 84].

In this study we designed the models to utilize the same amount of GPU memory during training disregarding the number of layers or parameters the models have. This design choice was made intentionally for several reasons: First of all, we wanted to allow each of the methods to play to their strengths. Second, we believe that setting an equal number of parameters or layers is not a sensitive constraint in the context of medical image segmentation. Most experiments are constrained by practical limitations, such as the GPU memory consumption, long before overfitting becomes a concern. It would certainly be interesting to see how the performance of the models changes with different constraints in place.

3.4. Discussion

In this Chapter we have conducted three case studies on semantic segmentation problems in the medical domain: brain tumor segmentation in multimodal MRI, cardiac substructure segmentation in cine MRI as well as kidney and kidney tumor segmentation in CT. We have developed state of the art algorithms for each of these problems and evaluated them on competitions to objectively and indisputably demonstrate their effectiveness. Our performance on the BraTS 2018 and KiTS 2019 challenges are hereby particularly noteworthy due to them being the most competitive and influential challenges in the domain: BraTS 2018 is the most recent iteration of BraTS, one if the oldest and influential competitions in the domain [38]. A total of 61 teams competed in 2018 among which were many high ranking institutions and industry players. KiTS

2019 was the most competitive challenge at the MICCAI conference 2019 with over 100 participating teams. The open leaderboard, even though it has only existed for less than a year now, has already over 700 submissions.

The segmentation problems treated in this chapter were quite different from each other, with each dataset having its own unique properties: With a typical size of $160 \times 190 \times 160$ and a spacing of $1 \times 1 \times 1$ mm, BraTS images are isotropic and nearly have cubic shape. Strong class imbalances, in particular for the enhancing tumor region can cause issues if not properly addressed. ACDC on the other hand is very anisotropic with a typical shape of $9 \times 256 \times 256$ and a corresponding spacing of $10 \times 1.56 \times 1.56$ mm impeding smooth transfer of information between slices. Slice misalignments further reduce the amount of useful information that can be transferred, causing 3D networks to not fully utilize the additional information. KiTS has a typical shape of $107 \times 512 \times 512$ with a corresponding spacing of $3 \times 0.78 \times 0.78$ mm and thus sits in between BraTS and ACDC in terms of anisotropy. Image sizes vary substantially, however, with the largest image being above $500 \times 500 \times 500$ voxels. Given memory limitations on modern GPUs trade off need to be made between image size and the retained granularity of details to enable the patch size to capture sufficient contextual as well as texture information.

The diversity of the datasets treated in this chapter and the solutions we came up with to address them teach valuable lessons about what design choices relate to good performance in a segmentation model. The perhaps most surprising result of this chapter is the strength of the plain U-Net. This architecture is very often used as a baseline and has been 'outperformed' numerous times. Yet, on all of the three datasets tested we used the standard U-Net formula to either define a new state of the art or closely match it. This observation is closely related to our discussion of the state of the art in medical image segmentation (see Section 2.5) where we hypothesized that the conclusions drawn in papers proposing new bells and whistles regarding network architecture may lack generality and a sufficiently thorough evaluation.

We believe that a key aspect contributing to the success of the U-Nets defined in this chapter lies in our dataset-specific adaptation. While all standard U-Nets we developed followed the basic concept of the architecture (i.e. encoder-decoder with skip connections, segmentation map generated at output stride 1, feature extraction with plain convolutions only (no residual, dense, ... connections)), we paid careful attention to the properties of the dataset and the need for adapting the network topology accordingly. In general, increasing the patch size (upper bounded by the image size) as much as the GPU memory allows provided great results on all tested datasets. Hereby, the patch size should be adapted to the anisotropy of the dataset, resulting for example in a patch size of $128 \times 128 \times 128$ for the isotopic BraTS and $10 \times 224 \times 224$ for the anisotropic ACDC dataset. Depending on the patch size and the spacing of the

corresponding dataset, the network topology needs to be appropriately adapted. All networks need to contain sufficient pooling operations such that the field of view of the encoder spans the entire input patch. This needs to be determined for each axis independently and sometimes pooling must be done for certain axes only in order to accommodate input dimensions with very low spatial size.

A part from the network topology, the experimental setting is an important variable that contributes to the success of failure of methods. In both BraTS and ACDC, not downsampling the images (by choosing the typical image spacing as resampling target) and thus training on the full image resolution provided the best results. In KiTS, however, the image data had to be downsampled in order to guarantee a proper field of view for the input patch size. Based on our observations on the ACDC dataset a 2D network may provide a better segmentation performance if the dataset is anisotropic and slice misalignments are present.

Dynamic, dataset-specific adaptations aside, we also observed several components of our pipelines to be consistent across all datasets, indicating that they constitute robust, well-performing design choices that can be considered generalizable across the diverse datasets encountered in the domain. These are for example the Dice loss function, extensive data augmentation, ensembling of models as well as resampling of training data to a common voxel spacing.

4. Automatic Design of Segmentation Pipelines

In the following chapter we will present nnU-Net, our framework for automating the design of segmentation methods in the biomedical domain. This chapter is based on the following publications ([30] and [23]):

Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P. F., Kohl, S., ... & Maier-Hein, K. H. (2018). nnu-net: Self-adapting framework for u-net-based medical image segmentation. arXiv preprint arXiv:1809.10486. `https://arxiv.org/abs/1809.10486`

Isensee, F.*, Jaeger, P. F.*, Kohl, S. A., Petersen, J., & Maier-Hein, K. H. (2020). Automated design of deep learning methods for biomedical image segmentation. arXiv preprint arXiv:1904.08128. `https://arxiv.org/abs/1904.08128`

(*: shared first authorship)

The two publications above describe different versions of the same method, nnU-Net, which is the key contribution described in the following chapter. While the initial version (first paper) was conceptualized, developed, implemented and evaluated by me, its further development (second paper) was done in collaboration with my co-author Paul F. Jaeger. Paul in particular contributed to discussions leading to the final version of the method. We furthermore collaborated on the interpretation of the results, a systematic analysis of the significance of this approach as well as its presentation. The final implementation of the approach, experiments and evaluation are my work.

4.1. Motivation

In order to achieve maximum performance, deep learning-based segmentation methods must be tediously adapted to the dataset they are dealing with, specifically targeting the properties and peculiarities of the associated segmentation problem. Considering the drastic variations in dataset properties encountered in the medical domain (see also Figure 1.2), it becomes evident that existing methods that were geared towards one specific dataset can not readily be transferred to arbitrary datasets or segmentation problems. It is important to emphasize that the success of a method depends on much more than the type of architecture that is being used. As we have seen in Chapter 3, state of the art performance can be achieved even with the standard U-Net architecture, but critically depends on appropriately selecting all the remaining hyperparameters under consideration of co-dependencies, hardware constraints as well as dataset-specific adaptations. Typical hyperparameters that need to be optimized are the target spacing for resampling, intensity normalization scheme, exact architecture topology (not architecture type), batch size, patch size, learning rate, momentum, data augmentation, postprocessing and many more. Among the segmentation pipelines we developed in Chapter 3, it is for example straightforward to see why the method developed for cardiac substructure segmentation (Section 3.2) cannot be used to segment kidneys and kidney tumors (Section 3.3): The data preprocessing is inappropriate, with the target spacing of the cardiac dataset being far too anisotropic for the kidney and kidney tumor segmentation task (KiTS), and the z-score intensity normalization technique being suboptimal for processing CT images. Furthermore, due to the importance of aggregating information across axial slices, a 2D network is unlikely to perform well on the KiTS dataset and the patch size selected in the 3D network does not consider sufficiently many slices. Since KiTS requires a different patch size for training a 3D U-Net, the topology of the architecture would also need to be adapted so that its receptive field properly covers the entire input. As a result, the memory footprint may change, requiring an adaptation of the batch size, which in turn can affect the learning rate required for optimal model training. While this example only covers the most obvious incompatibilities that can be identified at first glance, many of the other design choices regarding the training and inference scheme may also not transfer well to the KiTS task. In order to successfully transfer this method to KiTS, one would thus have to exhaustively identify and re-tune all the incompatible aspects of the cardiac substructure segmentation method, a process that needs expertise, time and compute resources. Intuitively, going through this entire process seems wasteful, error prone and it unlikely to result in optimal segmentation performance on the target task.

Surprisingly, this process plays a fundamental role in the current literature landscape

where new models are proposed by demonstrating superior performance relative to some baseline algorithm. These baselines were seldomly developed on the same dataset as newly proposed methods. In order to enable a meaningful comparison, authors are forced to manually adapt the baseline algorithm and re-tune them to their dataset at hand, often resulting in suboptimal baseline performance. To ease this process, authors often reduce baseline methods to their network architecture, such as the U-Net [43, 44] or V-Net [25], strip away all the other parts of the method and then proceed to utilize their own set of hyperparameters for the entire pipeline. With the resulting lack of a performance guarantee, it becomes increasingly difficult to discern whether newly proposed methods really constitute a veritable and long lasting improvement or whether they simply outperform a weak baseline. The availability of a standardized dataset-agnostic state of the art baselines would enable researchers to make more credible claims and make it easier for the reader to discern the value of the proposed method. Baselines aside, the dataset dependency has other far reaching consequences for the field of medical image segmentation. When developing new segmentation methods, authors generally avoid going through the effort of manual tuning for multiple datasets and therefore only use one (type of) dataset to demonstrate its effectiveness. While this may not be an issue with niche methods that specifically target certain requirements of one particular dataset, general methodological claims (such as 'a residual U-Net beats a plain U-Net' or 'our new loss function is better than the Dice loss') can simply not be made in this environment. This issue is amplified by the small size of datasets encountered in the biomedical domain. Even the largest datasets only have several hundreds of training cases [15, 35, 29, 45, 38], resulting in the potential for overfitting when only a limited number of datasets are being used. The small size of the datasets also causes large variations in performance when training the same method multiple times while varying the random seed, an issue that is not yet properly addressed in the literature. If newly proposed methods could be evaluated on multiple datasets without tedious manual re-tuning, both of these issues could be addressed, enabling researchers to better identify good design concepts and provide convincing proof thereof. Finally, the current situation has detrimental effects on users and researchers from other domains that require working segmentations algorithms for their work. They may not have the expertise to go through the manual adaptation process and thus do not get access to state of the art segmentation methods.

It is surprising that the dataset dependency of segmentation methods has so far not been identified as a major problem that is inhibiting the field from moving forward effectively. We hypothesize that this may be due to a persisting belief that specialized solutions are required for each of the diverse datasets and that a one-fits-all solution simply cannot exist or, if it does, not deliver acceptable performance. We strongly disagree with this assessment and will show as part of this chapter that segmentation

methods can indeed be designed without the dataset dependency in place and at the same time deliver or even exceed state of the art performance across a large variety of different datasets.

In this chapter we present nnU-Net, our framework for automated segmentation method design for biomedical image segmentation. It breaks the dataset dependency of conventional segmentation methods and thus directly addresses the aforementioned issues. In order to achieve this, we need to distance ourselves from the traditional fixed hyperparameter setting of segmentation methods and instead define nnU-Net as a method template that is dynamically and automatically adapted to each new dataset it encounters. The core idea behind nnU-Net is to dissect the many codependent hyperparameter choices that make up a segmentation pipeline into three categories: the *blueprint parameters* which remain constant for each of the datasets, the *inferred parameters* which encode those parts of the pipeline that need to be dynamically adapted to cope with the specific requirements of the dataset and finally the *empirical parameters* which are the ones that cannot be determined a priori and must be learned from data. Hereby, the experience gained from developing state of the art segmentation methods, such as the ones presented in Chapter 3, will be leveraged for assigning each hyperparameter to one of these groups and determining how they should be set.

4.2. Method

The principle behind nnU-Net is to redefine segmentation methods as dynamic templates that are automatically adapted to new datasets. We hereby explicitly revolve around the standard U-Net architecture as our network architecture type in an attempt to demonstrate how effective it can be when hyperparameters are selected appropriately. This design choice, which stands in stark contrast with the overwhelming focus of recent publication on searching better network architectures, coined the name of our framework: "No New Net" = nnU-Net. Note that this does not mean that nnU-Net is bound to use the U-Net. Any segmentation architecture can be integrated into the framework and adapted in a similar fashion.

In the following we first provide a broad overview of how nnU-Net works relative to traditional model design, followed by a deeper dive into its components. The differences between traditional method design and nnU-Net are highlighted in Figure 4.1. As shown in a), when developing a new segmentation method or adapting an existing one to a new dataset, researchers need to go through an iterative loop of manual method adaptation, changing the network architecture, preprocessing, training scheme, etc. while monitoring performance on a held-out validation set. This process is repeated until the performance is deemed satisfactory by the expert. Finally, the resulting model

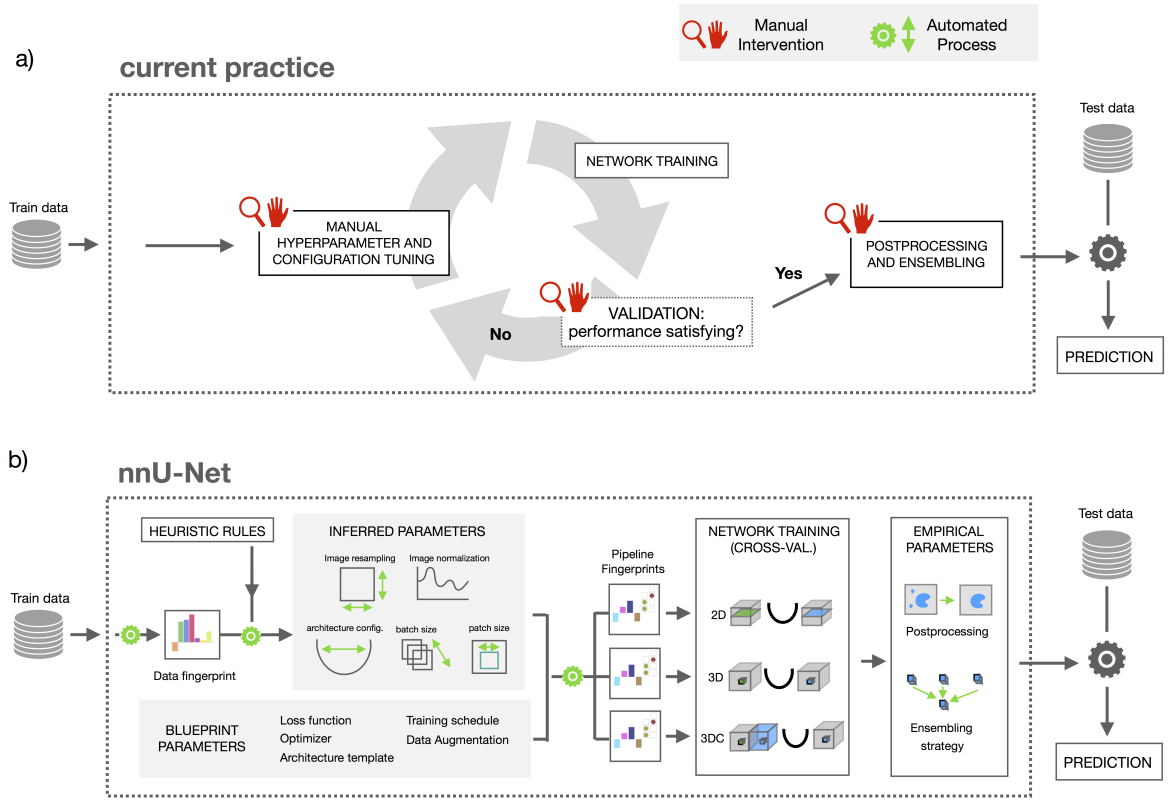

Figure 4.1.: **Manual vs. automated Method design.** a) highlights the iterative trial and error process required to manually identify a good segmentation model. It represents the entire optimization process, from preprocessing, over the exact hyperparameters used for training, all the way to the selected network architecture and postprocessing. This iterative process is non-standardized, seldomly documented in the associated publications and can cause wildly varying results depending on the expertise of the researcher. b) gives a condensed overview of the systematic approach taken by nnU-Net. First, a dataset fingerprint is extracted. Together with nnU-Nets empirical and inferred parameters, this fingerprint is then used to generated three separate segmentation pipelines. These are then trained in a cross-validation on the training cases. Finally, the best (ensemble of) configuration(s) as well as appropriate postprocessing are determined. Figure reproduced from [23].

can be applied to the test set. As we have discussed before, this form of model development is non-standardized, rarely documented in the associated publications and the resulting model performance largely depends on the expertise and skill of the researcher. nnU-Net on the other hand approaches the design process systematically (as shown in b)). First, the dataset is analyzed with respect to its key properties, such as number of training cases, image sizes and voxel spacings to create what we call a dataset fingerprint. nnU-Net then configures three segmentation pipelines (represented by pipeline fingerprints), each with a different type of U-Net-base architecture at its core: a 2D U-Net that operates slice-by-slice, a 3D U-Net that operates on full image resolution and a cascade of two 3D U-Nets where the first U-Net generates segmentations at a low resolution which are then refined by the second, full resolution U-Net. The previously mentioned *blueprint parameters* hereby define the fixed (i.e. not adapted) parts of the pipelines, such as the loss function, training hyperparameters and architecture template. The *inferred parameters* encode how nnU-Net utilizes the dataset fingerprint to make dataset-specific adaptations to preprocessing, network topologies, batch sizes and patch sizes used for training. Each pipeline is then trained as a five-fold cross-validation on the training cases of the dataset. Finally, nnU-Net automatically determines the best method or ensemble of methods as well as an appropriate postprocessing technique. The final output of nnU-Net are fully trained state of the art segmentation models that can be deployed to make predictions on unseen images.

In the following we describe the various parts of nnU-Net in detail. The overarching design principles should hereby also be interpreted as our best practice recommendations for developing and adapting segmentation methods.

4.2.1. Dataset fingerprint extraction

The very first processing step done by nnU-Net is to crop all images to the central nonzero region. The reasoning behind this step is that some datasets, typically those with segmentation tasks in the brain (see also Section 3.1) may contain large zero-valued areas around the region of interest. Excluding these non-informative areas from the following steps reduces the computational burden without sacrificing performance.

nnU-Net then extracts the dataset fingerprint, which is essentially a collection of the dataset properties that are required for the dataset-specific adaptation of segmentation pipelines. Specifically, nnU-Net collects all image sizes (i.e. number of voxels per spatial dimension) before and after cropping, voxel spacings (i.e. the physical size each voxel represents, typically measured in mm), the type and number of image modalities, the number of training cases and the number of classes that need to be segmented. For each input modality that is labeled as CT, nnU-Net furthermore computes the mean,

standard deviation as well as the 0.5 and 99.5 percentiles of the intensity values found in any of the foreground classes across all training cases.

4.2.2. Blueprint parameters

Blueprint parameters are the parts of the nnU-Net pipeline that remain fixed and are not adapted between datasets. These are use of a U-Net-like architecture template, the training schedule, data augmentation as well as the way inference is implemented. Note that while the architecture template is always the same, the actual topology of the network is heavily influenced by the inferred parameters. It may furthermore seem counter intuitive to use fixed settings for the training scheme and also in particular for the data augmentation. Empirical evidence suggests, however, that these do not need to be adapted to obtain good performance.

4.2.2.1. Architecture template

As we have shown in the previous chapter, state of the art segmentation results can be achieved on a range of datasets by properly adapting the standard U-Net architecture appropriately. Thus, all networks generated by nnU-Net follow the same, U-Net [44, 43] inspired template. We emphasize the use of the word *template* in this context because, while our network architecture follows the same design patterns as the U-Net, the implementation is dynamic and can be adapted to match the requirements of each dataset (see *Adaptation of Network topology, patch size, batch size* below). Specifically, all networks make use of the encoder-decoder pattern with skip connections and output stride 1. Notably, we do not use any of the recently proposed architectural variations such as residual connections [25, 9, 84], dense connections [93, 94], attention modules [97], dilated convolutions [77, 34] or squeeze and excitation modules [95, 96]. We make only minor necessary changes to the original U-Net formula. Motivated by the success of our manually tuned segmentation pipelines, nnU-Net enables large patch sizes at the cost of a smaller batch size. As we will see below, hardware limitations dictate that most of our 3D U-Nets only operate with a batch size of 2. In this context, batch normalization [139], which is commonly used to stabilize or speed up the training, does not perform well due to unreliable batch statistics [145, 182]. We therefore replace it with instance normalization [137] in all U-Net models. We furthermore observed an increase in training stability by replacing the standard ReLU nonlinearity with leaky ReLUs (negative slope 0.01) [138]. Motivated by our models from Sections 3.1.2 and 3.3, all networks are trained with deep supervision by adding additional segmentation heads and by applying corresponding losses to all but the two lowest resolutions. Gradient-based training causes the network to learn the simplest decision rules it can make to successfully solve the segmentation task on the training cases. Thus, if possible,

networks could end up using only the low level features from the upper layers in the U-shape. The resulting decisions, while possibly appropriate for the training set may, however, not generalize to the validation set and the test case. Our auxiliary losses inject gradients deeper into the network, forcing it to use all its layers and preventing it from bypassing the deep U-shape and making suboptimal decisions. All networks configure by nnU-Net use the common configuration of two blocks per resolution in both the encoder and decoder, with each block consisting of a convolution, followed by instance normalization and a leaky ReLU nonlinearity. Inspired by the concept of representational bottlenecks in [8], we implement downsampling by increasing the output stride of the first convolution in each resolution step of the encoder. Upsampling in the decoder is implemented by a convolution transposed. To strike a balance between representational power and GPU memory consumption, all networks are initialized with 32 feature maps at the highest resolution. This number is doubled (halved) with each downsampling (upsampling). Depending on the input patch size, the network can have up to 7 down- and upsampling operations (sometimes even more) which would result in $32 * 2^7 = 4096$ feature in the bottleneck. To prevent such an explosion in feature representations, the associated computational cost and unreasonable number of parameters, we do not allow the number of features to exceed 320 for 3D and 512 for 2D networks.

4.2.2.2. Training schedule

Based on the experience gathered in chapter 3, longer training schedules typically resulted in better training performance. Therefore, we let all U-Nets train for a total of 1000 epochs, with each epoch being defined as 250 iterations. Although we have repeatedly used the Adam optimizer in previous sections (see Sections 3.1 and 3.2), we observed in KiTS (Section 3.3) that better results can be obtained by swapping it out with stochastic gradient descent, an observation that is also in line with the literature [183]. All networks are thus trained with stochastic gradient descent with an initial learning rate of 0.01 and nesterov momentum ($\mu = 0.99$) as optimizer. The learning rate is decayed over the course of the training by applying the following formula at the end of each epoch: $lr = (1 - ep/ep_{max})^{0.9}$ [77], where ep is the current epoch, ep_{max} is the total number of epochs (here 1000) and lr is the current learning rate.

Even though the Dice loss specifically handles any class imbalance present in the dataset, it can only do so when voxels of the foreground classes are present in the patches. For patches with background only, the nominator of the Dice loss is 0, thus resulting in a missing penalty for false positive detections. Therefore, several measures are taken to facilitate the learning process. First of all, the Dice loss [24] is complemented with the categorical cross-entropy loss, which is effective even if only

background is present and can thus suppress false positive detection. Furthermore, we employ oversampling to increase the proportion of the patches containing foreground classes: During training, minibatch are constructed by first randomly selecting training cases from the current training set. 33.3% of training samples are guaranteed to contain at least one of the foreground classes. This is implemented by randomly selecting one of the foreground classes that is present in the corresponding case and then cropping the patch around a randomly chosen voxel belonging to this class. The remaining 66.7% of patches are selected from random locations. We furthermore identify networks where the image size is expected to be much larger than the patch size (for example all 2D U-Nets and the 3D full resolution U-Net if a 3D low resolution U-Net is present) and change the way the Dice loss is computed for these cases. Typically, the Dice loss would be computed for each sample of the minibatch independently ('sample Dice') and then averaged over the samples to obtain the overall value of the loss. As discussed before, this can result in false positives not being punished properly in samples that only contain background voxels. Therefore, we adapt the Dice loss computation in the affected networks so that all samples in the minibatch are treated jointly as a large pseudo-volume ('batch Dice'). With the oversampling in place, this approach guarantees the presence of foreground voxels and thus a better optimization. We should note that the lack of false positive penalization could also have been achieved by explicitly optimizing the background class with the Dice loss, or by adding a constant value to both the nominator and denominator of the Dice loss. Empirically, however, the proposed approach of Dice and cross-entropy loss, oversampling and 'batch Dice' (if appropriate) resulted in overall superior segmentation performance.

The loss is computed independently for each of the segmentation heads of the network. The auxiliary segmentation outputs at lower resolution are herefore paired with correspondingly downsampled versions of the reference annotation. The overall training objective is the weighted sum $L_{\text{total}} = w_1 * L_1 + w_2 * L_2 + w_3 * L_3 + ...$ of the losses L_i computed at different resolutions (with higher i denoting lower resolution). Due to the lower spatial extent of the auxiliary segmentation heads their respective losses are down weighted the closer to the bottom of the U-shape they are located, resulting in $w_2 = \frac{1}{2}w_1; w_3 = \frac{1}{4}w_1$ etc. The weights of the losses are normalized so that they sum to 1.

As we have seen in our previously configured segmentation methods, data augmentation is pivotal to enable good generalization even when using only a limited number of training cases. We therefore apply a broad variety of data augmentations on the fly during training: scaling, rotation, Gaussian noise, Gaussian blur, contrast, brightness, gamma, low resolution simulation and mirroring along all spatial axes. Note that each of the augmentation is associated with a probability of being applied. All augmenta-

tions are implemented with our publicly available *batchgenerators* toolkit[1]. Please refer to the Appendix A.1 for an exhaustive description of the data augmentation techniques used.

4.2.2.3. Inference

Images are predicted with a sliding window approach. Hereby, the network is slided across the image and processes windows of the same size as the patch size used during training. Adjacent predictions overlap by half the patch size. Predictions are merged by averaging the softmax outputs of the network across all predictions. Due to the padding of convolutions, the accuracy decreases towards the edges of the individual predictions. To prevent stitching artifacts, we use a Gaussian importance weighting that gives higher weights to the center voxels during softmax aggregation. We furthermore apply test time augmentation in the form of mirroring for a slight increase in segmentation accuracy.

The use of a sliding window approach over fully convolutional prediction is motivated by potential GPU memory constraints with large CT images as well as the use of instance normalization. Although we have used fully convolutional inference with this type of normalization before (Section 3.3), this approach yields bad segmentation accuracy when the patch size is very different from the image size and can therefore not be applied universally in a framework that is designed to process arbitrary datasets. The underlying reason for this lies in the zero padding of convolutions: during training, the feature maps in the lower layers of the U-shape will have a narrow spatial extent (all the way down to only 4 voxels, see below). Here, padding the feature maps such that subsequent convolutions produce representations of the same size results in a large proportion of zeros creeping in from the sides. During training, these zeros substantially influence the normalization of feature maps through instance normalization (which computes the mean and variance of each feature map) and the network weights are trained with the resulting shift in intensity distribution in mind. If the input size during inference is then substantially larger than the patch size used for training, the relative proportion of zero or near zero-valued voxels is lower, resulting in a different output of the normalization and thus incompatibility with the trained network weights.

4.2.3. Inferred parameters

The inferred parameters establish a direct link between the dataset fingerprint and large parts of the pipeline that is generated by nnU-Net. They are effectively determined by a set of heuristic rules which encode our expert knowledge on how segmentation

[1] `https://github.com/MIC-DKFZ/batchgenerators`

methods must be designed in order to be effective. These rules reflect our expertise in the domain, which stems from many years of experience, multiple winning contributions in segmentation competitions (as described for example in Sections 3.1.3, 3.2 and 3.3) as well as the design of high impact clinical pipelines (see Section 3.1.2).

4.2.3.1. Intensity Normalization

Intensity normalization is an essential preprocessing step aiming at reducing heterogeneities in the data that could impede the learning process. The default normalization scheme is used for all imaging modalities except CT. Hereby, each image is normalized individually by subtracting their mean and dividing by their standard deviation. If the cropping resulted in an average size decrease of 25% or more on the training cases, nnU-Net creates a nonzero mask and applies the normalization to the voxels within this mask only, leaving the remaining voxels at zero. The default scheme is extremely robust and has proven its value in previous chapters: it effectively reproduces the normalization used for brain tumor segmentation (Section 3.1) and cardiac substructure segmentation (Section 3.2).

As we have already discussed in Section 3.3, CT intensity values encode how much radiation is absorbed at a specific spatial location, thus representing physical properties of the tissue. Therefore, unlike in MRI, voxel intensities are standardized between scanners, allowing us to leverage a level-window like approach which clips the image intensities to the range of interest. This approach is similar to the way clinicians interact with these images [184]. In clinical practice, appropriate windowing is determined by looking up standardized values for the organ of interest. Since this information is not accessible to nnU-Net, it determines the windowing directly from the training cases: The previously computed intensity statistics within the foreground region are used for this purpose (see 4.2.1): Image intensities are clipped to the 0.5 and 99.5 percentiles and then normalized by subtracting the mean and dividing by the standard deviation.

4.2.3.2. Resampling

Datasets in the biomedical domain are rarely homogeneous with respect to the voxel spacing of the individual images. The voxel spacing encodes how much space in the physical world each voxel represents. CNNs, and thus also all networks designed by nnU-Net operate on voxel grids only and cannot represent or interpret this property. Ignoring the voxel spacing altogether would result in the sizes, aspect ratios and textures of the objects of interest to be non-representative of their real-world distribution. While having variation in the data can be beneficial to some degree, for example by varying the scales and aspect ratios slightly via data augmentation, unrealistic and uncontrolled variations make it substantially harder for the networks to detect the objects

properly, resulting in overfitting and poor test set performance. Therefore, there is a great interest in homogenizing the voxel spacings within a dataset.

For images, nnU-Nets default interpolation method is third order spline interpolation. It deviates from this method only for anisotropic data (defined as maximum axis spacing $> 3 \times$ minimum axis spacing). Anisotropic spacings can cause large variations between slices which often result in interpolation artifacts with third order spline interpolation. To suppress these artifacts, nearest neighbor interpolation is used for interpolation along the out-of-plane axis (in-plane resampling is still done with spline interpolation). Segmentations are resampled by first converting them to a one hot encoding. Then each class is interpolated separately by linear interpolation. Finally, the segmentation map is recovered with the argmax operation. For anisotropic images, the out of plane axis is again resampled with nearest neighbor interpolation.

4.2.3.3. Target spacing

We have already touched on the importance of the target spacing for resampling in Section 3.3.3.1 in the context of the KiTS challenge. The target spacing controls how large the objects of interest are (in voxels). Selecting a low target spacing results in high resolution images and large objects whereas choosing a large target spacing results in low resolution image and small objects. Higher image resolutions retain more texture information, but with given hardware limitations, make it difficult to configure patch sizes which would allow for sufficient contextual information to be collected. Low image resolutions enable the network to potentially see all the context they need, but suffer from poor detail in the remaining texture. Furthermore, the segmentations generated at low resolution are lacking fine grained details that cannot be recovered. When designing a segmentation method, one must find a careful balance between contextual information on the one hand and the retained texture information and fine structures on the other. If domain knowledge is available, the decision can be swayed in either direction depending on the properties of the segmentation task. In a framework such as nnU-Net, dataset-specific domain knowledge is not available and we must rely on empirical evidence to select the proper target spacing. This is the reason for configuring two separate 3D U-Net pipelines: The full resolution 3D U-Net and the U-Net cascade.

The full resolution 3D U-Net, as the name implies, operates at the original image resolution of the dataset. It partly ignores the fact that it may potentially not collect sufficient contextual information (partly because nnU-net still attempts to make the patch size as large as possible, see 4.2.3.4). However, considering that the voxel spacings within a dataset are seldomly homogeneous, this is not as straightforward as it may appear. In some datasets, there can be large variations, sometimes even as large as a factor of 5 between the highest and lowest resolution images [35]. To find a proper

target spacing for this configuration, we assume that the typical image in the dataset will have a voxel spacing that is appropriate for the task at hand. Thus, per default, nnU-Net selects the median of all the voxel spacings found in the training cases as the target spacing. This is computed for each axis independently. Once again, anisotropic cases are treated differently. Due to potential large variations between slices, it is beneficial to select a lower target spacing (=higher resolution) for the out-of-plane axis. Here, anisotropy is defined as having both a voxel and spacing anisotropy > 3 (computed based on median spacing and median size in voxels). The target spacing for the out-of-plane axis of anisotropic datasets is selected as the 10th percentile, but is not allowed to deceed the in-plane spacing.

Similarly to the 3D full resolution U-Net, the 2D U-Net also processes the images at full resolution. Since it operates on in-plane slices only, resampling is also done only along those axes and the out-of-plane axis is left unchanged. The target spacing for the in-plane axes if hereby selected following the same principle as for the 3D full resolution U-Net by simply picking the median value found across all training cases for that axis. Note that the two-dimensional nature of the network enable nnU-Net to configure substantially larger patches, which is why we assume that the 2D network always covers sufficient contextual information and does not require a cascaded approach.

Note that the configuration of the 3D U-Net cascade is more involved because it is entangled with the configuration of the exact network topology. This process is described separately in Section 4.2.3.5.

4.2.3.4. Adaptation of Network topology, patch size, batch size

Considering the segmentation problems treated in Chapter 3 it quickly becomes clear that there exists no combination of network topology and patch size that could satisfy the requirements of arbitrary datasets. In BraTS (Section 3.1.3), for example, the image size was approximately isotropic with about $160 \times 190 \times 160$ being a typical image shape. This meant that a patch size of $128 \times 128 \times 128$ seemed like an appropriate choice: it covers a large part of the input and its cubic shape provides a receptive field that is equally large along all axes. In ACDC 3.2, the typical image shape is $9 \times 256 \times 216$. Its anisotropy must be reflected in the patch size used to segment it: $10 \times 224 \times 224$. Applying the BraTS patch size to this dataset would not make sense due to the excessive padding and wasted computation in the first axis. In KiTS (Section 3.15), the patch size ($80 \times 160 \times 160$) was quite a bit smaller than the image size after resampling ($128 \times 248 \times 248$) and followed the aspect ratio of the resampled images. Using the ACDC patch size for this dataset may have resulted in insufficient context aggregation along the first axis whereas the BraTS patch size would have overemphasized contextual information along this axis (note that the target spacing of

KiTS was anisotropic: $3.22 \times 1.62 \times 1.62$ in mm and the BraTS patch size was isotropic in voxels). For each of the datasets, the topology was carefully designed to optimally deal with the respective patch size.

Designing appropriate network topologies is a difficult problem that requires careful considerations. Given some desired input patch size, the networks must ensure that the receptive field at the bottleneck spans the entire input patch so that as much contextual information as possible can be extracted. Depending on the size of the patch, this may require a different number of pooling operations. Following our network template which dictates the use of two computational blocks (2× (convolution - instance norm - lReLU)) per resolution, this implicitly couples the network depth to the patch size - a sensible design choice given that larger patches are more difficult to aggregate and may require more expressive, and therefore deeper network architectures. The anisotropy of the patch size must furthermore be considered: if the spacing discrepancy between axes is too large, aggregation of information across slices may have harmful effects on the networks performance, as we discussed already for the ACDC dataset in Section 3.2.6. Furthermore, anisotropic axes may require less or sometimes even no pooling operations at all. Finally, the memory footprint of the network is one of the major constraints that must be integrated into the design process. Without it in place, we would simply set our patch size to match the image size in a dataset and then configure the architecture to match it. With it in place this is often not possible, and we need to resort to an iterative optimization scheme to design the network.

In the following we describe how network architectures for the 3D full resolution U-Nets (both cascaded and non-cascaded) and the 2D U-Net are designed in nnU-Net. The design of the low resolution stage of the cascade is described below in Section 4.2.3.5. The network architecture design requires the median image shape of the dataset after resampling as well as the target voxel spacing as input. Figure 4.2 a) provides an overview of the network design process.

Initialization

The patch size is initialized to the median image shape of the dataset. Since U-Net-like architectures require input patch sizes that are divisible by $2^{\text{num\_pool}}$ (where num_pool is the number of downsampling operations), the patch size is padded appropriately.

Architecture topology

nnU-Net then generates a network topology that optimally uses the given patch size. The topology follows the template defined in 4.2.2.1: U-Net-like encoder-decoder with skip connections and output stride 1, two computational blocks per resolution in both encoder and decoder. It is configured by determining the number of downsampling

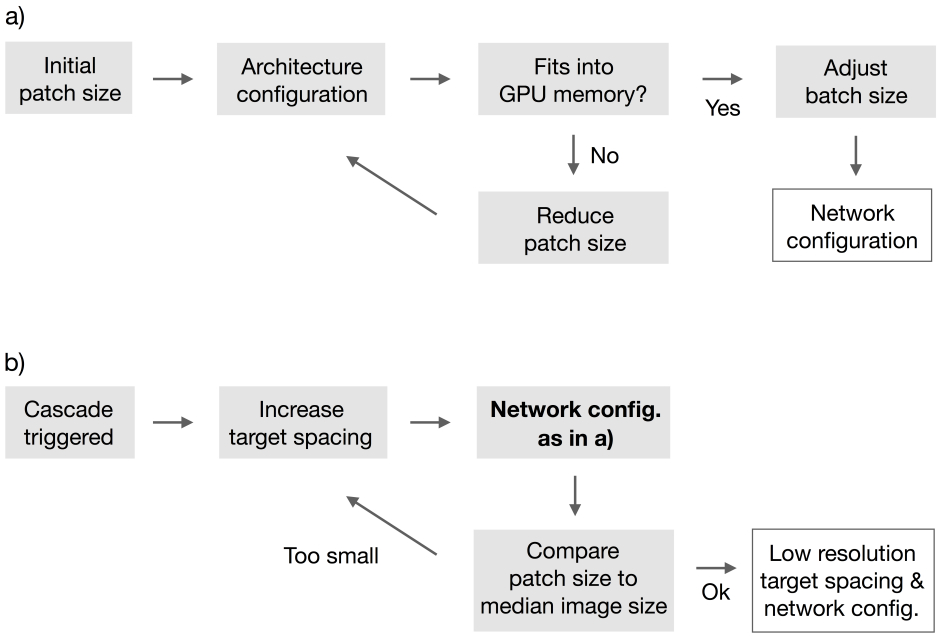

Figure 4.2.: **Iterative optimization of network topology and U-Net cascade.** a) shows how the network topology, patch size and batch size are determined depending on the median image shape and corresponding voxel spacing. b) demonstrates the configuration of the 3D low resolution U-Net configuration of the U-Net cascade. The target spacing is successively increased in an outer loop, thus reducing the image size. The patch size, topology and batch size are optimized in an inner loop following the same principle as in a). The outer loop is terminated once the patch size is larger than 25% of the median image shape. Figure reproduced from [23].

operations along each axis depending on the spacing and size of the input patch. To this end, the stride of the convolutions used for downsampling is set to 2 to aggregate contextual information along the high resolution axes. Once the spacing of all axes is within a factor of 2, downsampling is performed for all axes simultaneously. Downsampling is terminated when further downsampling would either result in a feature map size < 4 in the bottleneck or the feature maps have become anisotropic.

The default kernel size is $3 \times 3 \times 3$ and 3×3 for all feature map generating 3D and 2D convolutions, respectively. Based on our observations in the ACDC dataset 3.2, aggregation across slices with a spacing anisotropy should be avoided. Therefore, the convolutional kernel size is set to 1 for the out of plane axis if its spacing is $> 2\times$ the spacing of the other axes. Note that once the kernel size was set to be isotropic, it remains this way and cannot go back to use a size of 1 along any of the axes.

The kernel size and output stride of the convolution transposed in the decoder are set to match the stride of the corresponding downsampling operation in the encoder.

Adaptation to GPU memory budget

As we have stated before, the amount of GPU memory available is the major constraint when configuring a network architecture. Initially, most patch sizes configured in the *Initialization* will be far too large. Take, for example, the Liver and Liver Tumor dataset [35] where the typical image shape is $432 \times 512 \times 512$ (which is what would be used as initial patch size) and compare this with the maximum patch size that we could configure in the KiTS section: $80 \times 160 \times 160$.

In particular when considering nnU-Nets intended use as an out-of-the-box segmentation tool, it is important to keep the GPU requirements in check: most users will not have access to large GPU cluster or even single expensive datacenter-grade GPUs such as the Nvidia V100 (32GB) or recently announced A100 (40GB). In order to target a large audience, we configure nnU-Net to be compatible with regular, consumer grade GPU hardware: All networks nnU-Net configures should be guaranteed to run on a Nvidia RTX 2080ti with 11GB of GPU memory. That said, nnU-net can naturally also be configured to fill the additional space available on larger graphics cards, if desired.

Based on our experience in the previous chapter, we enforce a minimum batch size of 2. The smaller the batch size, the noisier the gradients the network will be trained with are going to be. We feel that a minimum batch size of 2 provides a good trade-off between the patch size that can be configured and training stability.

In order to adapt the patch size, and with it the network architecture, to match our hardware constraint, we first need to estimate what the GPU memory consumption is going to be. This is not a straightforward process, given that different convolution

algorithms may be selected by Nvidia's cuDNN library. Nonetheless, we empirically observed that the memory consumption largely and predictably correlates with the total number of voxels encountered in all feature maps in the network. This makes sense given that the largest proportion of the GPU memory must be used to store the feature map activations during the forward pass so that they can be used in the backward pass for gradient computation. The memory required for network parameters is negligible in comparison. Thus, we use the number of voxels of a manually configured topology with known GPU memory consumption as our reference value against which new topologies can be compared during their optimization.

Thus, once a network topology has been configured, we estimate its memory consumption with the aforementioned heuristic. If the network does not fit into the GPU with a batch size of at least 2, the patch size and thus the network topology must be adapted. To this extent, the axis that is proportionally largest relative to the median image shape is identified and its size is reduced by 2^n where n is the number of downsampling operations of that axis. Then, a new network topology is generated that is optimally adapted to the new patch size. This process, which is indicated by the circle 'Architecture configuration → Fits into GPU memory? → Reduce Patch size' in Figure 4.2 a), is repeated until the patch size is small enough.

Batch size

Especially in segmentation problems where batch sizes are often quite small, larger batch sizes substantially reduce the noise in the gradients used to train the network. If the initial patch size was small enough and the corresponding topology immediately passed the check for the GPU memory constraint, there may be additional headroom to increase the batch size used for training. The same heuristic that was also used in the previous paragraph is then used to increase the batch size until the entire GPU memory is used. Note that, in order to prevent overfitting, the batch size is capped so that the voxels contained in the minibatch do not exceed 5% of the total number of voxels in all training cases.

4.2.3.5. Configuration of 3D U-Net cascade

The 3D U-Net cascade is intended to alleviate issues arising from missing contextual information in the 3D full resolution U-Net if the images are very large. In a sense, it constitutes the best of both worlds: The 3D low resolution U-Net operates on a low spacing and thus sees all the context it needs. The resulting coarse segmentations are then refined by the second U-Net that operates at the same voxel spacing as the 3D full resolution U-Net. For this purpose, the upsampled coarse segmentations of the low resolution U-Net are concatenated to the high resolution image data and processed

by the network. Hereby, the coarse segmentations substitute the potentially missing contextual information while the network can extract additional fine grained structures and textural information directly from the image. The configuration of the 3D full resolution U-Net is identical to the non-cascaded 3D U-Net and is handled by the process described in Section 4.2.3.4. The configuration of the low resolution U-Net is more involved, because it must be designed such that the patch size covers a significant proportion of the input images to enable the collection of the contextual information. To achieve this, a separate downsampled version of the training data must be generated, the target spacing, and thus the resulting image shapes of which must be optimized jointly with the network topology.

The process of configuring the cascade is summarized in Figure 4.2 b). Note that the cascade is not configured for datasets where the 3D full resolution U-Net already covers a large proportion of the image size. Specifically, it is omitted if the patch size of the 3D full resolution U-Net exceeds 12.5% of the median image shape after resampling (indicated by 'Cascade triggered' in the Figure). If the cascade is triggered, nested optimization loops are used to identify a suitable configuration. In the outer loop, the target spacing for the low resolution data is successively adapted. It is initialized to be the same target spacing as is also used for the full resolution training. In each iteration, it is multiplied by 1.01. If the spacing is anisotropic, only the spacing of the high resolution axes is increased until all axes have a spacing within a factor 2. The resulting new median image shape (along with the spacing) are then used to configure a network topology using the process described in 4.2.3.4 (inner loop). The target spacing is successively increased until the configured patch size exceeds 25% of the current median image shape and the configuration of the low resolution U-Net is complete.

4.2.4. Empirical parameters

The pipeline parameters that cannot be estimated solely based on the dataset fingerprint are summarized in this chapter.

4.2.4.1. Model selection and ensembling

As discussed previously, nnU-Net generates up to three different U-Net pipelines, the 2D U-Net, the 3D full resolution U-Net and the 3D U-Net cascade. Each of these pipelines comes with its own advantages and disadvantages, and it may be difficult to predict which of them should be selected for which dataset (see also our discussion in Section 4.4). Thus, nnU-Net trains all configured pipelines in a five-fold cross-validation on the training cases and empirically selects the best performing configuration (or ensemble of configurations) based on the average Dice score. Inference is always done

using the five models stemming from the training set cross-validation. If an ensemble is selected, the five models of each ensemble member are utilized. Note that ensembles never exceed two configurations to limit the inference time.

4.2.4.2. Postprocessing

nnU-net furthermore empirically determines whether connected component analysis improves the results of the selected configuration(s). To this end, it uses the predictions of the five-fold cross-validation. It first treats all foreground classes as one and determines whether removing all but the largest connected component increases the average Dice score on the training cases. Then, it uses the result of this first stage to determine whether removing all but the largest component individually for each class further improves the result.

4.3. Results

nnU-Net overcomes the narrow definition of traditional segmentation methods, where all parts of the pipeline used to be manually tuned for a single dataset. Instead, nnU-Net provides a dynamic method template that is molded fully automatically to meet the requirements of any dataset in the domain. This is made possible by condensing expert knowledge about the design of segmentation pipelines into inductive biases which shortcut the high dimensional optimization problem that needed to be solved previously. We used the training set of the 10 datasets provided by the Medical Segmentation Decathlon (MSD)[2] [29] for the development of nnU-Net. Specifically, five-fold cross validation or single train-val splits were used to find all heuristic rules found in the *inferred parameters*, optimize the fixed *blueprint parameters* and identify the concepts behind the *empirical parameters*. These 10 datasets provided sufficient variability to ensure generalization to other, previously unseen datasets in the domain.

In the following we demonstrate that, despite its automated nature, nnU-Net achieves state of the art performance across a variety of datasets without requiring any manual intervention. Naturally, our evaluation only includes datasets stemming from international segmentation competitions, allowing us to compare our segmentation performance against the respective state of the art on each of the datasets. We furthermore show how nnU-Net can be used for method development and evaluation on multiple datasets and why this evaluation scheme should be used for robust decision making.

4.3.1. nnU-Net handles a variety of datasets and image properties

We apply nnU-Net to the 10 datasets originating from the Medical Segmentation Decathlon. To truly test its generalization to unseen tasks, we identified 9 additional

[2] `http://medicaldecathlon.com/`

segmentation competitions that provide one dataset each for a total of 19 diverse datasets spanning 49 different segmentation tasks. These 19 datasets are the same as the ones presented in Figure 1.2 which highlights their diversity by extracting the respective dataset fingerprints. For each of the datasets, nnU-Net is applied without manual intervention: we simply let nnU-Net analyze the datasets, configure and train its pipelines and finally choose which models and postprocessing it should use. The resulting configuration was then applied to the holdout test sets. Note that for some datasets, in order to ensure proper stratification of training data, we manually interfered in the data splits used for cross-validation (details are disclosed in the Appendix A.2).

Figure 4.3 shows qualitative segmentation results of nnU-Net on 12 different datasets. All examples shown originate from their respective test set. In each example, an overlay of the raw image data with the generated segmentation is shown to the left and a 3D rendering highlighting the tree dimensional nature of the segmentation tasks is shown to the right. As can be seen in the Figure, nnU-Net handles all segmentations effortlessly, generating high fidelity and accurate delineations. In particular, the diversity of the imaging modalities used for evaluation is highlighted: CT images (a, d, e, h, i, l), different types of MRI (c, f, g, j, k), multi-modal MRI (c, j) as well as serial section Transmission Electron Microscopy (b).

4.3.2. nnU-Net outperforms specialized, manually tuned state of the art pipelines

Figure 4.4 provides an overview of the quantitative results achieved by nnU-Net. Each of the 49 segmentation tasks is displayed separately. nnU-Net is show as a red dot while all competing methods are shown in blue. Even though nnU-Net is a generic segmentation method that needs to adapt itself to each of the 19 different datasets, it was able to outperform all competitors on 29 out of the 49 segmentation tasks. In the remaining 20 segmentation tasks, nnU-Net is very competitive with scores close to the top of the leaderboard. This is a remarkable result given that the competing methods were all hand tuned by experts to the respective datasets at hand.

4.3.3. nnU-Net designs appropriate segmentation pipelines

ACDC dataset

In order to demonstrate the nnU-Net indeed generates appropriate segmentation pipelines we picked two example datasets, ACDC [4] and LiTS [35], and discuss the choices made by nnU-Net.

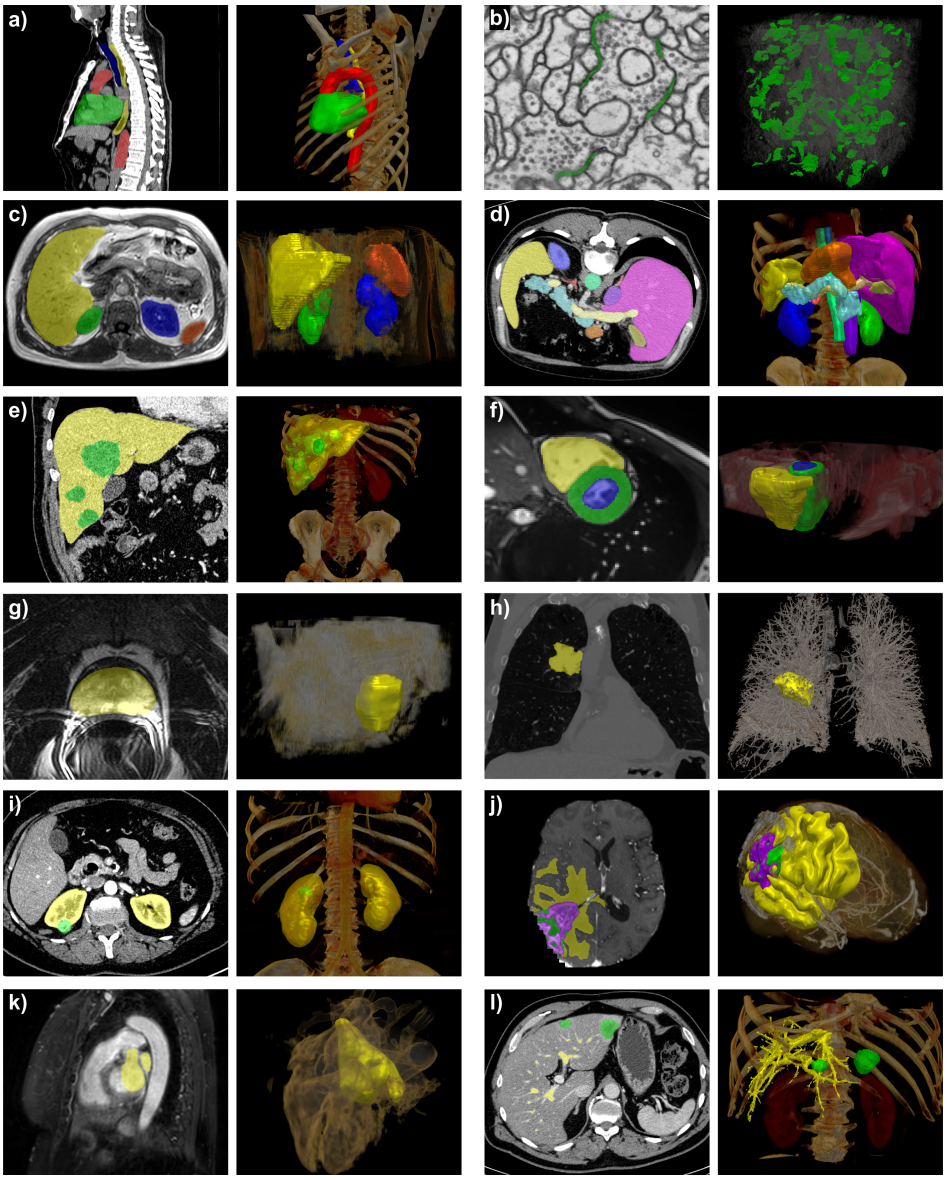

Figure 4.3.: **nnU-Net handles a broad variety of segmentation tasks, input modalities and dataset properties.** Examples originate from the test set of their respective dataset. left: overlay of nnU-Net's segmentation with raw data, right: volume rendering (generated with MITK [181]). a: heart (green), aorta (red), trachea (blue) and esophagus (yellow) in CT images [185]. b: synaptic clefts (green) in electron microscopy scans (https://cremi.org/). c: liver (yellow), spleen (orange), left/right kidney (blue/green) in T1 in-phase MRI [186]. d: thirteen abdominal organs in CT images [36]. e: liver (yellow) and liver tumors (green) in CT images [35]. f: right ventricle (yellow), left ventricular cavity (blue) myocardium of left ventricle (green) in cine MRI [4]. g: prostate (yellow) in T2 MRI [42]. h: lung nodules (yellow) in CT images [29]. i: kidneys (yellow) and kidney tumors (green) in CT images [15]. j: edema (yellow), enhancing tumor (purple), necrosis (green) in MRI [29]. k:left ventricle (yellow) in MRI [29]. l: hepatic vessels (yellow) and liver tumors [29]. Figure reproduced from [23].

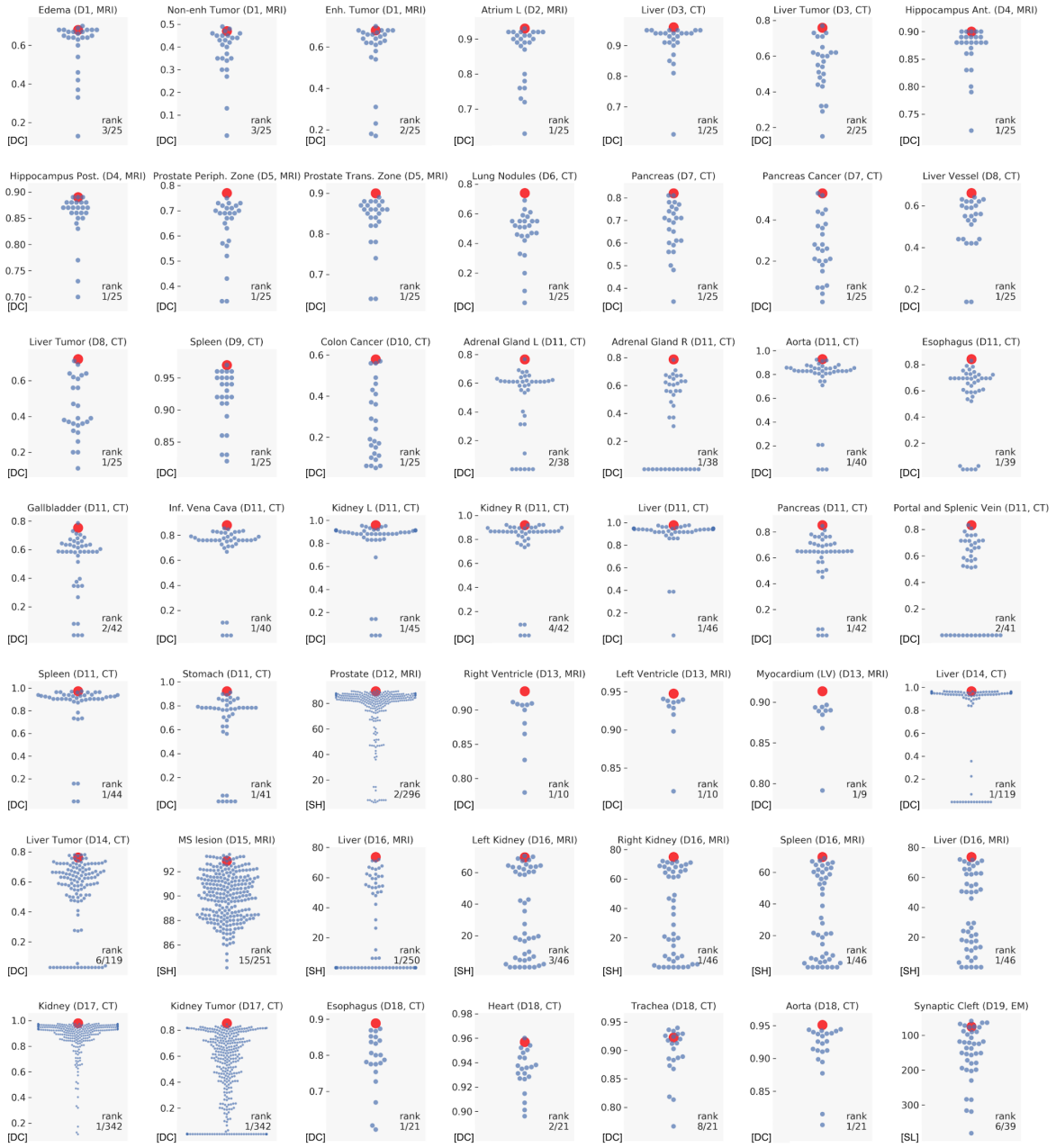

Figure 4.4.: **nnU-Net outperforms manually designed segmentation pipelines on a variety of segmentation tasks.** We tested nnU-Net by applying it to 19 diverse datasets spanning 49 different segmentation tasks. Each task is plotted separately. Competing methods are shown in blue, nnU-Net's result in red. Overall, nnU-Net sets a new state of the art on 29 out of the 49 testes segmentation tasks. DC: Dice score. SL: Score (lower is better). SH: Score (higher is better). Figure reproduced from [23].

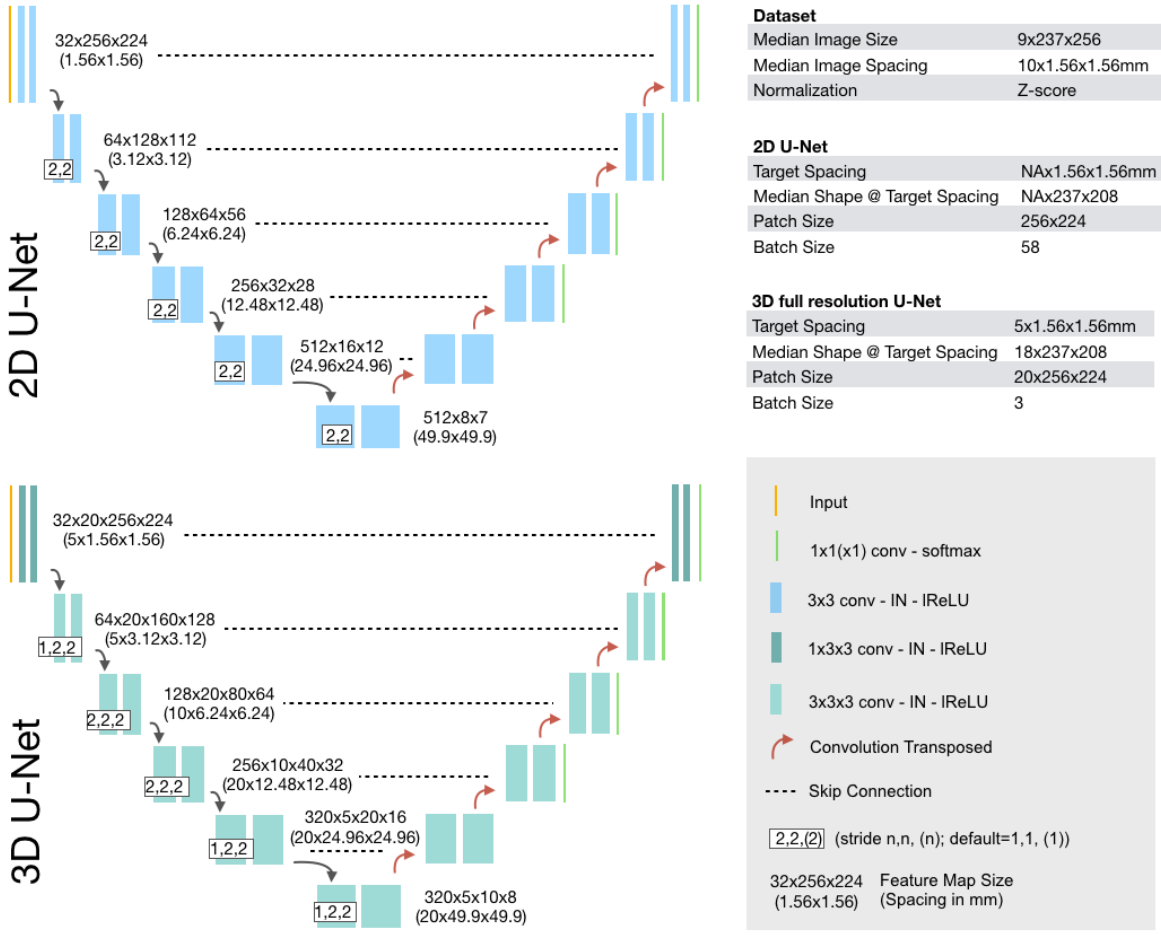

Figure 4.5.: **Network topologies generated by nnU-Net for the ACDC dataset.** The 2D U-Net is shown at the top and the 3D full resolution U-Net at the bottom. The U-Net cascade was not configured for this dataset because the patch size of the full resolution 3D U-Net already covers entire images. Key dataset properties, target spacings as well as corresponding median image shapes are shown to the right. Figure reproduced from [23].

Figure 4.5 summarizes the network architectures generated by nnU-Net for the ACDC dataset [4]. With the modality being cine MRI, nnU-Nets default intensity normalization scheme is used. Thus, all images are normalized independently by subtracting their mean and dividing with their standard deviation. The target spacing for in-plane resampling is selected as the median value found in the training cases: 1.56mm. Due to the anisotropic nature of the images, the out of plane target spacing is selected as the 10th percentile and is thus set to 5mm for the 3D full resolution U-Net (instead of 10mm, which would be the median). The 2D U-Net processes patches of size 256×224 with a batch size of 58. It uses 5 pooling operations, resulting in a feature map size in the bottleneck of 8×7, guaranteeing a sufficiently large receptive field for good segmentation performance. The 3D full resolution U-Net processes patches of size $20 \times 256 \times 224$. At the selected target spacing, the median image shape of the training cases is $18 \times 237 \times 208$. When comparing median image size and patch size, while also considering that the feature map size in the bottleneck is just $5 \times 10 \times 8$ it becomes evident that this network topology has a receptive field that virtually sees entire training cases at once and can thus make optimal segmentation decisions. Note that the 3D full resolution U-Net uses $1 \times 3 \times 3$ kernels for the convolutions at the highest resolution to prevent aggregation of information across slices, a design choice that was missing in our manually tuned approach (Section 3.2) and may have caused suboptimal performance. Another critical improvement of nnU-Net is the target spacing for the out-of-plane axis. While we manually selected 10mm in Section 3.2.3.1, a choice that caused interpolation artifacts on all cases with a lower voxel spacing, nnU-Net selects 5mm target spacing. The 3D U-Net cascade is not configured for this dataset.

	RV	MLV	LVC	mean
2D	0.9053	0.8991	0.9433	0.9159
3D_fullres	0.9059	0.9022	0.9458	0.9179
Ensemble	0.9145	0.9059	0.9479	0.9227
Postprocessed	0.9145	0.9059	0.9479	0.9228
Test set	0.9295	0.9183	0.9407	0.9295

Table 4.1.: **ACDC results.** All reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. *Postprocessed* denotes the cross-validation Dice scores of the ensemble after applying nnU-Nets postprocessing. The Dice scores for the test set are computed with the online platform. The online platform reports the Dice scores for end diastolic and end systolic time points separately. We averaged these values for a more condensed presentation. Table reproduced from [23]

The superiority of nnU-Nets approach regarding the 3D full resolution becomes clear

when analysing the cross-validation results reported in table 4.1. Compared to our manually tuned algorithm, which achieved an average Dice score of only 0.8930 for the 3D U-Net (see table 3.4), the 3D U-Net configured by nnU-Net has substantially higher cross-validation performance with an average Dice score of 0.9179. nnU-Nets 2D U-Net was approximately on par with our previous, manually designed method with a score of 0.9159 (versus 0.9173). The better performance of the 3D full resolution U-Net is also reflected in the test set results where the ensemble selected by nnU-Net achieved an average Dice of 0.9295 which is marginally higher than our challenge winning contribution, which achieved 0.9275 with a very similar approach.

LiTS dataset

The LiTS dataset [35] consists of CT images with a median image size of $482 \times 512 \times 512$ after resampling (3D full resolution configuration). It has a high, nearly isotropic resolution with a median spacing of $1 \times 0.77 \times 0.77$ mm in the training cases. The nnU-Net-generated pipelines for this dataset are summarized in Figure 4.6. Due to the image modality being CT, nnU-Net utilizes the global normalization scheme based on the foreground voxel statistics computed across the training cases (see Section 4.2.1). Each image is normalized by first clipping to $[-17, 201]$ and then normalizing with mean 99.4 and standard deviation 39.39.

The 3D full resolution U-Net pipeline uses the aforementioned median spacing of the training cases as target spacing for resampling. Its patch size of $128 \times 128 \times 128$ reflects the voxel and spacing isotropy of the dataset. However, due to the size of the images, this network only sees $\frac{1}{60}$ of the image at a time. Therefore, nnU-Net also configures the 3D U-Net cascade for this dataset. As described previously, the full resolution part of the cascade uses the same settings as the 3D full resolution U-Net, but takes the segmentations generated by the low resolution U-Net as additional guidance by concatenating them to its input. As can be seen in the Figure, the 3D low resolution U-Net operates on images that were downsampled to $2.47 \times 1.90 \times 1.90$ mm, resulting in a median image shape of $195 \times 207 \times 207$. At this resolution, the input size of the low resolution U-Net ($128 \times 128 \times 128$) correctly covers 25% of the median image shape (as described in Section 4.2.3.5) and can thus collect sufficient contextual information. Note that the network topology and input patch size of the 3D low resolution U-Net are only coincidentally identical to its full resolution counterpart. On datasets with anisotropic voxel spacing, low and full resolution U-Nets will have a different topology.

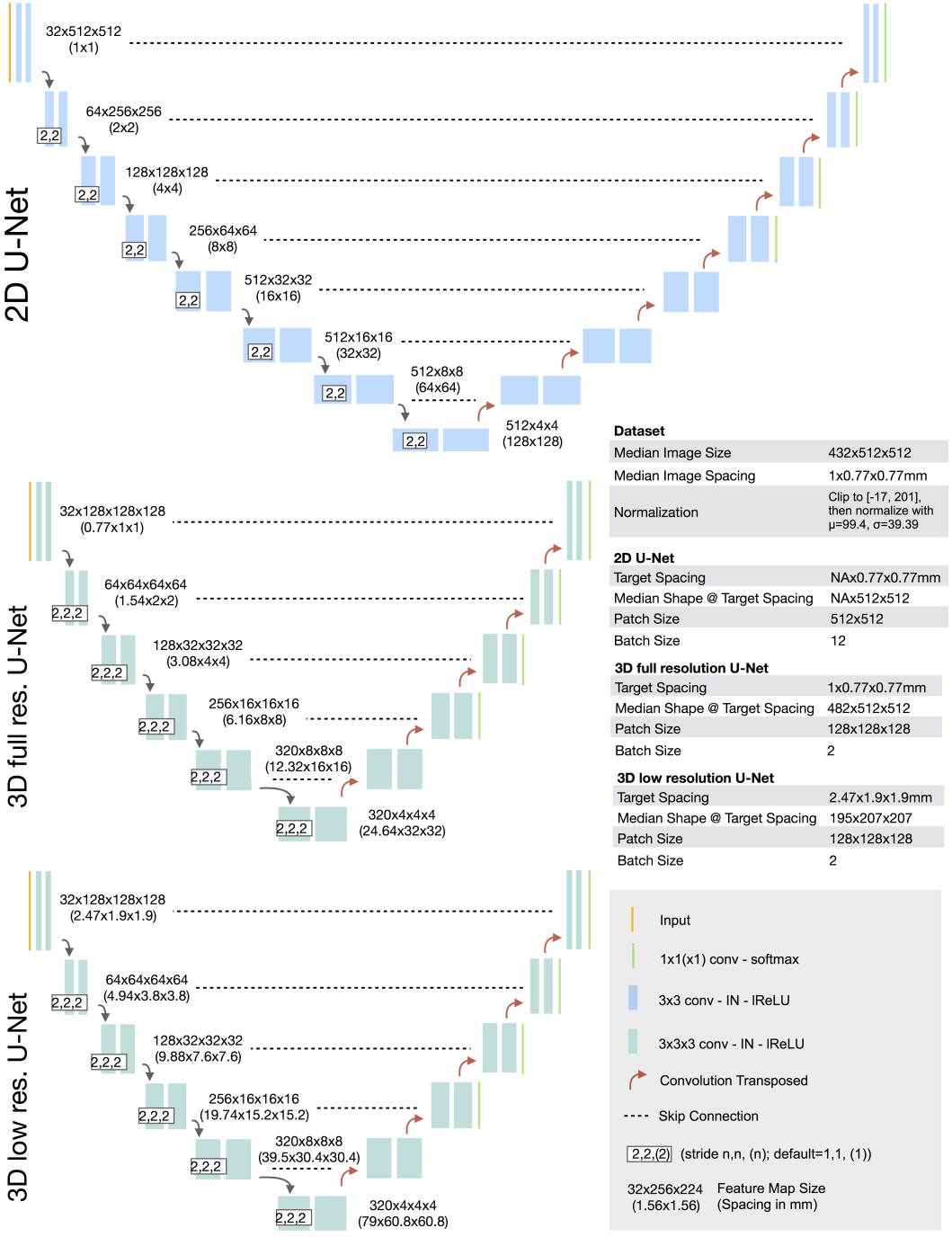

Figure 4.6.: **Network topologies generated by nnU-Net for the LiTS dataset.** The 2D U-Net is shown at the top, the 3D full resolution U-Net in the middle and the 3D low resolution U-Ne at the bottom. The input patch size of the 3D full resolution is much smaller than the median image shape U-Net, causing the U-Net cascade to be triggered. Key dataset properties, target spacings as well as corresponding median image shapes are shown to the right. Figure reproduced from [23].

	liver	cancer	mean
2D	0.9547	0.5603	0.7575
3D_fullres	0.9576	0.6253	0.7914
3D_lowres	0.9585	0.6161	0.7873
3D cascade	0.9609	0.6294	0.7951
Best Ensemble*	0.9618	0.6539	0.8078
Postprocessed	0.9631	0.6543	0.8087
Test set	0.9670	0.7630	0.8650

Table 4.2.: **LiTS results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. Best ensemble on this dataset was the combination of the 3D low resolution U-Net and the 3D full resolution U-Net. *Postprocessed* marks the Dice scores after nnU-nets postprocessing is applied to the ensemble. Note that the Dice scores for the test set are computed with the online platform. Table reproduced from [23].

nnU-Nets results on the LiTS dataset are summarized in Table 4.2. Due to inter-slice information being not accessible to it, the 2D U-Net is by far the worst performing pipeline. Interestingly, despite the missing contextual information, the 3D full resolution U-Net achieved an average score of 0.7914, which is not far off the 3D U-Net cascade (0.7951). This could indicate that the availability of contextual information is not essential for solving this segmentation problem. Among the 3D results, the 3D low resolution performed the worse with 0.7873, which is most likely due to the coarseness of the generated segmentations (note that all evaluations are done on the original voxel spacing of the images, requiring the generated segmentations to be resampled). The best ensemble for this dataset was the combination of the low and full resolution U-Net, which provides a substantial improvement over either of the single model configurations with an average score of 0.8078 before and 0.8087 after postprocessing. It is unclear whether one should draw any conclusions from the model selection for the ensemble since the second best combination, the ensemble of the 3D full resolution U-Net and the cascade was not far off with a score of 0.8086. On the test set, nnU-Net achieved Dice scores of 0.9670 for the Liver (first place) and 0.7630 for the Tumor class (rank 6 out of 119). While it would be interesting to compare nnU-Nets results and methodology with other leaderboard entries, the LiTS leaderboard [3] unfortunately neither links its entries to corresponding publications nor does it enforce comprehensible usernames. Furthermore, the use of external data is specifically allowed, potentially contaminating the leaderboard with entries benefiting from it and making it difficult to discern methodological from data advantages. When compared to the original challenge win-

[3] `https://competitions.codalab.org/competitions/17094#results`

ner of the 2017 challenge [26], who achieved liver and tumor scores of 0.961 and 0.722 (without using external data), respectively, nnU-Net has far superior performance.

A comprehensible summary of all challenge participations and generated segmentation pipelines is provided in the Appendix A.2.

4.3.4. Evaluation across multiple datasets enables more robust design choices

Throughout this manuscript we have repeatedly stated that evaluation of new segmentation methods should be done on as many datasets as possible to avoid overfitting and to increase the credibility of methodological claims. In the medical domain in particular, the small size of the datasets can result in substantial noise in the reported Dice scores and, if not handled properly, result in the wrong conclusions being drawn. Here we illustrate how nnU-Net can be used to explore the value of different methodological variants addressing different parts of the pipeline. We implement them into nnU-Net and then use its generic nature to evaluate them via 5-fold cross-validation on the 10 training datasets of the Medical Segmentation Decathlon [29]. It should be emphasized that *implement once - evaluate on many datasets* was impossible previously: authors had to manually redesign and tune appropriate segmentation pipelines for each dataset they evaluate on. Given the complexity of this process, authors regularly did not go through the effort causing most recently published methods to be evaluated on just a single (type of) dataset.

The variants we exemplarily selected are: two alternative loss functions (plain Cross-entropy (CE) and TopK10, a variant of CE in which only the worst 10% of the predictions are used for gradient computation [187], the introduction of residual connections [9] in the encoder of all generated U-Nets, using three convolutions per resolution in both encoder and decoder instead of two (resulting in a deeper network architecture), two modifications of the optimizer (replacing SGD with Adam [140] and using a smaller momentum term of 0.95 instead of nnU-Nets 0.99), replacing instance normalization [137] with batch normalization [139] and removing all data augmentation. Through their integration into nnU-Net, these variants only need to be implemented once, but can still be tested on an arbitrary number of datasets. Here we configured all of them to use the same GPU memory constraint as nnU-Net's base model to ensure a realistic and fair comparison.

Figure 4.7 shows the results of applying these variations to the 10 MSD datasets. The bars represent the distribution of rankings across bootstrap samples, first for each dataset separately and finally aggregated over all datasets. These volatility of the rankings underline the danger of evaluating new methods on a small number of datasets: While five out of the 9 tested variants achieved the highest rank in at least one of the datasets, none of them was able to consistently outperform the nnU-Net baseline.

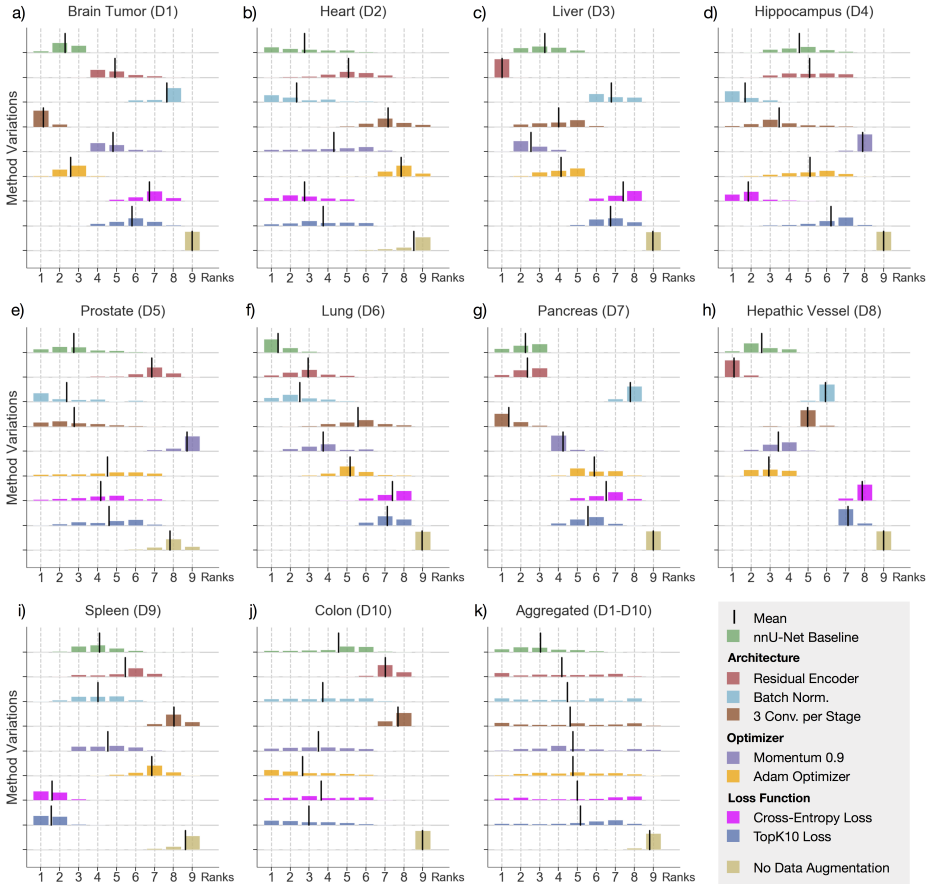

Figure 4.7.: **Evaluation across multiple datasets reduces noise and enables more robust design choices.** We implemented nine different pipeline variations into nnU-Net and evaluated them across the ten Medical Segmentation Decathlon datasets against nnU-Nets baseline configuration. The results are shown both independently for each dataset (a-j) as well as aggregated over all datasets (k). The distribution over ranks is generated by generating 1000 virtual validation sets via bootstrapping. Even though five of the nine tested variants achieved the first rank in one of the datasets, none of them could consistently outperform nnU-Nets baseline. Figure reproduced from [23].

Looking forward, we would like to see generalizing frameworks such as nnU-Net to be rigorously used for method development. nnU-Net's dynamic nature requires researchers to implement their changes only once while enabling them to evaluate on an arbitrary number of datasets. Hereby, the default nnU-Net configuration should be used as a robust, dataset-agnostic state of the art baseline.

4.3.5. nnU-Net is freely available as an out-of-the-box tool

nnU-Net is open source and freely available at GitHub (`https://github.com/MIC-DKFZ/nnUNet`). It can also be installed via the Python Package Index (PyPi). Please refer to the nnU-Net GitHub page for an extensive documentation on how to use it. The source code also comprises a large number of nnU-Net variations, including the ones used in Section 4.3.4, which can be used as starting points for learning how to modify nnU-Net. We furthermore make pretrained models for all datasets used in this Chapter available for download. They are accessible at `https://zenodo.org/record/3734294` or via the nnU-Net command line interface.

4.4. Discussion

In this Chapter we have presented nnU-Net, the first generalizable out-of-the-box segmentation framework. It takes away with the narrow hard coded definition of traditional segmentation methods and instead formulates a dynamic pipeline template that is automatically adapted to the properties of each dataset at hand, thus breaking for the first time the dataset dependency that held back methodological research in the domain.

Despite its generic nature, nnU-Net was able to outperform manually designed and highly optimized methods on a broad range of segmentation challenges, setting a new state of the art in the majority of segmentation tasks it was tested on. This is particularly noteworthy because there previously seemed to be a belief that handcrafted, specially designed segmentation methods are required to achieve state of the art performance.

Instead of going through the complex, high dimensional optimization problem that is traditional segmentation method development for each of the datasets, nnU-Net uses domain knowledge in the form of inductive biases to shortcut the process: Its careful selection of fixed *blueprint parameters*, dataset-dependent dynamic adaptations with the *inferred parameters* as well as data driven *empirical parameters* enables it to narrow down the vast search space of possible methods to just three configurations: a 2D U-Net, a 3D U-Net that operates on full resolution images as well as a 3D U-Net cascade.

Perhaps surprisingly, the strong performance of nnU-Net was achieved by intelligently adapting the standard U-Net architecture and combining it with well-optimized and proven concepts. In a literature landscape that focuses so heavily on finding elaborate architectural variants, these results raise important questions about the validity of the evaluation that are being performed when proposing new concepts. Despite the small datasets size in our domain, authors still demonstrate methodological improvements only on a single (type of) dataset and using non-standardized baselines. In Section 4.3.4, we have demonstrated how this process is inherently flawed and may result in the wrong conclusions being drawn. To overcome this problem, we propose to use nnU-Nets dynamic method template for model development, allowing authors to roll out their changes to arbitrarily many datasets. The standard nnU-Net should hereby be used as standardized state of the art baseline.

As highlighted in the qualitative as well as quantitative results (Sections 4.3.1 and 4.3.2), the segmentations generated by nnU-Net are highly accurate. Since we require no user intervention, nnU-Net is the first segmentation tool that can truly be used out of the box on any dataset in the biomedical domain. It requires reasonable compute resources: anyone with a standard Nvidia 2080ti graphics card or better now has access to the state of the art in semantic segmentation. This empowers users who do not have the expertise, time or compute resources to do model development themselves.

Previous research done in the field of Automated Machine Learning (AutoML) addressed similar questions to nnU-Net: how can we find good machine learning methods with as little human intervention as possible [101, 100]? The problem, however, was so far approached from a different angle.
Existing research in AutoML predominantly attempts to empirically find the very best *specialized* method for a given dataset. These methods are mostly data-driven and directly optimize some validation metric. As a result, they share the same disadvantages as traditional manually designed algorithms: their output is a fixed set of hyperparameters that is optimal only for the one dataset they were optimized for. When they need to be applied to a new dataset, the optimization process needs to start all over again, entailing large requirements with respect to compute resources. nnU-Net on the other hand focuses on maximal *generalization* and keeps the amount of empirical decisions to a bare minimum. This is achieved by our unique combination of blueprint, inferred and empirical parameters. nnU-Net is the culmination of years of experience in designing state of the art semantic segmentation methods combined with a development process that explicitly included multiple datasets for finding and validating all design choices. This allowed us to identify which parameters need changing when confronted with a new dataset and which parameters can remain constant (blueprint parameters). Among the parameters that need changing, we were able to identify underlying con-

cepts which can be used to derive the overwhelming majority of them automatically at virtually no computational cost (inferred parameters), leaving solely the model selection and postprocessing as our empirical parameters.
Essentially, speaking in terms of bias-variance trade-off, nnU-Net is highly biased and uses strong regularization in the form of explicitly implemented domain knowledge for its decision making. Compared to previous AutoML-based approaches, the number of hyperparameters that need to be optimized in nnU-Net are purposefully minimal. This allows nnU-Net to design entire segmentation pipelines while remaining within a realistic computational budget and preventing it from overfitting to the dataset at hand. While existing research in AutoML also encodes expert knowledge to some extent, in particular through search space restrictions and search heuristics, their larger number of empirically determined parameters naturally translates into a higher variance and thus a more pronounced tendency to overfit. Particularly in the medical domain, where datasets are comparatively small, this can be detrimental.
To the best of our knowledge, there exists no other AutoML-based method that is able to optimize the pipeline in its entirety, from preprocessing over the network architecture, training scheme all the way to postprocessing. One likely reason for this is that the search space that would need to be defined for the corresponding optimization is difficult to identify even for single parts of segmentation pipelines. Attempting to optimize the pipeline as a whole, and then on top of that making this optimization compatible with the diverse datasets encountered in medical image segmentation appears insurmountably complex and would require compute resources that only the very largest institutions could afford. Thus, unlike nnU-Net, there also exists no tool based empirical AutoML approaches that can be downloaded and used by anyone to achieve state of the art segmentations. But even when moving the focus away from the design of the entire pipeline and concentrating on AutoML methods that only optimize specific parts of the pipeline it becomes clear that these methods are still not ready for real world applicability. We conducted a thorough analysis of the leaderboards of current segmentation challenges and found that traditional AutoML-based methods are surprisingly absent. Specifically, we have analyzed all 100 entries in the KiTS competition[4] [28] as well as the winning contributions to multiple other competitions [35, 45, 38, 188, 186]. With the sole exception being the 18th place of KiTS [5] (which uses simple grid search for hyperparameter optimization), we have not been able to find any other AutoML-based approaches. It is difficult to say why that is. We hypothesize that a combination of several factors could be contributing to their absence: as discussed above, the compute requirements for current empirical AutoML-based methods are still quite large and may discourage participants to use them in the particularly

[4] http://results.kits-challenge.org/miccai2019/
[5] http://results.kits-challenge.org/miccai2019/manuscripts/peekaboo_2.pdf

computationally intensive task that is 3D image segmentation. Furthermore, with at best only a couple hundred training cases, datasets in medical image segmentation are fairly small in comparison with datasets in classical computer vision (CIFAR-10 has 50.000 training examples and ImageNet over one million), making overfitting a real problem. nnU-Net on the other hand has been applied successfully to a large number of competitions spanning 19 diverse datasets in the domain, underlining its real world impact and usefulness as an out-of-the-box segmentation tool.
That said, we do not intend to disregard the usefulness of empirical AutoML-based methods by any stretch of the imagination. Just because right now, possibly due to the limited size of datasets available for automated model design, a convincingly successful application to medical image segmentation is missing, this will certainly not continue to be the case. With the availability of larger datasets as well as improved computational efficiency of empirical AutoML-based methods in the future, we expect their impact to increase substantially. Essentially, we see nnU-Net as orthogonal to empirical AutoML-based methods. As pointed out previously, empirical AutoML methods excel at optimizing certain parts of a pipeline, making them a prime candidate for replacing certain parts of the nnU-Net pipeline. We can very well imagine Neural Architecture Search [111, 117, 116] taking over the network architecture design or methods like AutoAugment [189] improve upon nnU-Nets fixed data augmentation.

While nnU-Net shows exceptionally strong performance across the 49 tested segmentation tasks, it has its limitations. Right now, its decisions are made purely based on the dataset fingerprint as well as limited data-driven experiments. When confronted with a segmentation task that needs a different evaluation metric than the Dice score, or can only be solved by incorporating dataset-specific expert knowledge into the pipeline it may struggle to deliver competitive performance. We have seen this on the synaptic cleft segmentation task of the CREMI dataset (`https://cremi.org`). While nnU-Net delivered competitive results (rank 6/39), electron-microscopy-specific preprocessing, data augmentation as well as a specialized loss function appear to be necessary to surpass the state of the art [190]. In highly domain-specific cases such as this one, nnU-Net should be seen as a strong starting point for making necessary modifications.

Despite its strong performance, we merely consider nnU-Net a first step in the direction of fully flexible and automatic segmentation frameworks. There are multiple ways on how it could be improved. For example, some pipeline parameters, such as the data augmentation, are currently part of the *blueprint parameters*. While this works reasonably well, one could attempt to derive dataset-dependent adaptations to, for example, the data augmentation to improve the performance even further. Also, as we have seen for the CREMI dataset, some modalities might require different preprocessing techniques. For recurring cases, one could therefore consider including additional

heuristics to specifically address them.

5. Discussion

Semantic segmentation is one of the most researched tasks in medical image computing. Numerous new methods are proposed each year, complemented by a thriving landscape of segmentation competitions enabling objective comparison of methods in a standardized environment.

In Chapter 3 we have taken a close look at three fundamentally different segmentation problems and designed specialized methods with highly competitive segmentation performance.
In Section 3.1 we have developed two separate algorithms for brain tumor segmentation in multimodal MRI. Brain tumor segmentation is characterized by isotropic image spacings, a stark class imbalance, uncertainty in the expert annotations as well as ambiguous and difficult to discern tumor regions.
First, in a more clinically motivated setup (Section 3.1.2), we developed a U-Net with residual connections in the encoder, a lightweight decoder and deep supervision. This network was trained on a large in-house dataset and evaluated on a large scale multi-institutional cohort, where we were able to show that our model has good generalization and radiologist-level segmentation performance. More importantly, we were able to demonstrate that tumor progression analysis based on our volumetric segmentation maps was significantly more robust than the clinical state of the art, which consists of manually drawn perpendicular diameters and a set of heuristics. The method developed in this project was furthermore evaluated on the BraTS 2017 competition in which it obtained the third place out of 47 competing methods.
Second, in Section 3.1.3 we approached brain tumor segmentation from a different angle: instead of using an elaborate network architecture, we purposefully restricted ourselves to use a standard U-Net-like network and attempted to maximise segmentation performance by optimizing the remaining parts of the pipeline. Surprisingly,

this method based on a plain architecture obtained the second place out of over 60 competitors.
In Section 3.2 we investigated cardiac substructure segmentation in cine MRI. These images are notoriously anisotropic, with the in-plane spacing being multiple times lower than the out-of-plane spacing. This poses several challenges in network design: aggregation of information across slices can introduce errors and negatively impact the performance of segmentation networks, but is ultimately required to make optimal segmentation decisions. Again we make use of the standard U-Net architecture: Our proposed approach was an ensemble of standard 2D and 3D U-Nets. Even though the 3D U-Net was vastly outperformed by its 2D counterpart on our five-fold cross-validation we still observed a small gain in the most difficult region, the right ventricle, when the two methods were combined. This highlights the importance of inter-slice information even if it causes a drop in segmentation accuracy when used in a standalone model. Our approach was evaluated in the ACDC segmentation competition where it outperformed all competing methods and won the challenge.
Finally, in Section 3.3 we tackled kidney and kidney tumor segmentation in CT images. These images are much larger than the maximal input patch size that can be processed under realistic hardware constraints. This required us to downsample the original images to effectively increase the receptive field of the networks. Since downsampling comes at the price of less fine-grained texture information as well as coarser segmentations, the target spacing had to be selected carefully. We furthermore took the large training dataset as an opportunity to make an attempt at outperforming the standard 3D U-Net architecture by designing two counterparts that make use of residual connections in their encoders. Even though one of the residual variants ended up being the best performing method (and thus selected for test set prediction) the difference to the 3D U-Net was marginal, preventing us from declaring a clear winner. Our segmentation method was evaluated as part of the highly competitive KiTS challenge where it outperformed over 100 competing methods and won the competition. We are confident that the plain 3D U-Net, had we selected it for test set prediction, would also have achieved the first place of the challenge.

Considering the current literature landscape, it appears surprising that we were able to surpass or match state of the art segmentation performance with standard architectures embedded in well-tuned pipelines on multiple different segmentation tasks and challenges. This raises important questions about the current state of research where new segmentation methods focusing on finding elaborate network architectures [95, 97, 191, 94, 34], loss functions [192, 193, 194, 195], pretraining schemes [196] and even neural architecture search [116, 197] are regularly proposed. Despite the overwhelming effort that is put into designing these methods, none of them constitute a necessary condition for good performance in segmentation challenges: We have re-

peatedly outperformed competitors by utilizing the 'baseline', a well tuned 3D U-Net. This dichotomy is quite interesting: How can methods, when they are proposed, be demonstrated to outperform some baseline and then proceed to be outperformed by it when tested in a standardized environment where proper tuning effort was put into it? Why does the review process of newly proposed papers not require a, or ideally multiple, successful challenge participation? And why are there no standardized, properly tuned baselines against which the proposed methods can be compared? The current research landscape indeed appears like an impenetrable jungle of methodologies where it becomes increasingly difficult to discern which of the methods really constitute a veritable and long lasting improvement.

In the discussion of the state of the art (Section 2.5) as well as the motivation in Chapter 4, we hypothesize that the underlying cause of this replication crisis lies in the combination of a high dataset diversity (see Figure 1.2) with a strong dataset dependency of segmentation methods. Traditional segmentation methods have a fixed set of hyperparameters, pre- and postprocessing scheme as well as network architecture, causing them to be either incompatible or not performing well on other datasets than the one they were optimized for. This is detrimental when attempting to adapt existing state of the art and baseline methods to the dataset one is currently working with, a process that is inconsistent, error prone and ultimately results in an underperforming reference which in turn makes it easier to propose a new, 'better' method. It really cannot be overstated how important evaluation in a standardized challenge environment, or, if unavailable, the use of a standardized baseline is! We should, however, not stop there. It seems to be common practice to evaluate new design concepts on only a single dataset or type of dataset (such as two different liver segmentation datasets). Yet, despite the narrow problem-specific evidence, authors readily make generalizing claims about their method. We have highlighted in Section 4.3.4 why conclusions being drawn from such a setup should be treated with suspicion: the small size of the datasets in our domain causes a substantial amount of noise in the results and may cause suboptimal configurations to sporadically perform better than a more generalizing and robust baseline. When running experiments for any of the proposed segmentation methods in this thesis, we observed variations in average Dice scores of up to 5% depending on the dataset. We therefore conclude that the ideal setup for proposing new methods should be to participate in as many competitions as possible, collecting indisputable evidence for the value of the proposed methods by outperforming other highly optimized segmentation methods on the respective datasets. If suitable competitions are absent, the least authors should do is choose a standardized baseline that is configured very carefully to give good performance.

As a side-effect of the dataset dependency of traditional segmentation methods, there

exists no out-of-the-box tool with which non-experts and researchers from other domains could get access to state of the art segmentation performance for their custom datasets. This constitutes a substantial problem, in particular in the medical domain where new unique datasets are regularly created to address new segmentation problems or address existing segmentation problems with different imaging modalities. These datasets are often created by clinicians who may not have experts at their disposal that are experienced enough to make the best out of the dataset. Even if experts are available, there is often no incentive for them to spend a lot of time on this type of task: Developing a solid segmentation method for a new dataset is unrewarding, as the resulting method rarely provides sufficient novelty to qualify for publication.

In this thesis, specifically in Chapter 4, we have taken a first step towards breaking the dataset dependency of segmentation methods. Our proposed framework, nnU-Net, removes the barriers imposed by a rigid method definition and instead provides a method template that is molded to each dataset it is applied to. This process is fully automatic and requires neither expert knowledge nor user interaction. It is made possible by transforming the domain knowledge gathered in developing the segmentation methods in Chapter 3 into inductive biases that shortcut the high-dimensional optimization process that is segmentation method development in the biomedical domain. nnU-Net uses nothing more but well-adapted and tuned standard U-Net architectures in its designs. We evaluated nnU-Net in the harshest possible environment by competing in 19 diverse datasets originating from 10 different international segmentation competitions. On each of the respective datasets, we competed against the best of the best: manually tuned algorithms that were hand-crafted to optimally solve the segmentation task at hand. Despite it's generic nature and fully automated application to these datasets, nnU-Net was not only able to compete, but in fact set a new state of the art on the majority of segmentation tasks. These results highlight how careful method development under consideration of multiple datasets as a collective training set enabled us to overcome the noisiness of the results and make more robust design choices.

While its fully automated nature qualifies nnU-Net as the ideal out-of-the-box segmentation tool that makes state-of-the-art segmentation available to experts and non-experts alike, we see its most impactful contribution in the way it will enable segmentation method development looking forward. Not only can it act as a standardized baseline against which newly proposed models should be compared regardless of the dataset the evaluation is being done. Crucially, nnU-Net is a framework into which changes can be incorporated easily. Researchers can benefit from it by using it to implement their methodological variant and exploit its dynamic method template to run evaluation of said variant on an arbitrary number of datasets.

We expect nnU-Net to substantially impact the way method development will be done

looking forward. It is already being adopted by numerous researchers in the domain who either use it as a baseline or as a framework for developing new concepts. We make nnU-Net available to the community as an open source tool and are excited to see how it will be improved in the future.

List of Own Publications

The following list only contains selected publications. It is sorted by citations as of June 22nd 2020. For a full list of publications, please refer to `https://scholar.google.com/citations?user=PjerEe4AAAAJ&hl=en`.

Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., ..., **Isensee F.**, ... & Prastawa, M. (2018). Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. arXiv preprint arXiv:1811.02629.

Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P. A., ..., **Isensee F.**, ... & Sanroma, G. (2018). Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved?. IEEE transactions on medical imaging, 37(11), 2514-2525.

Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., & Maier-Hein, K. H. (2017, September). Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge. In International MICCAI Brainlesion Workshop (pp. 287-297). Springer, Cham.

Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., & Maier-Hein, K. H. (2018, September). No new-net. In International MICCAI Brainlesion Workshop (pp. 234-244). Springer, Cham.

Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P. F., Kohl, S., ... & Maier-Hein, K. H. (2018). nnu-net: Self-adapting framework for u-net-based medical image segmentation. arXiv preprint arXiv:1809.10486.

Isensee, F.*, Jaeger, P. F.*, Full, P. M., Wolf, I., Engelhardt, S., & Maier-Hein, K. H. (2018). Automatic Cardiac Disease Assessment on cine-MRI via Time-Series Segmentation and Domain Specific Features. Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges (pp. 120–129). Cham: Springer International Publishing. `https://doi.org/10.1007/978-3-319-75541-0_13`

Isensee, F.*, Jaeger, P. F.*, Kohl, S. A., Petersen, J., & Maier-Hein, K. H. (2019). Automated design of deep learning methods for biomedical image segmentation. arXiv preprint arXiv:1904.08128.

Kickingereder, P.*, **Isensee, F.***, Tursunova, I., Petersen, J., Neuberger, U., Bonekamp, D., ... & Maier-Hein, K. H. (2019). Automated quantitative tumour response assessment of MRI in neuro-oncology with artificial neural networks: a multicentre, retrospective study. The Lancet Oncology, 20(5), 728-740.

Isensee, F., Kickingereder, P., Bonekamp, D., Bendszus, M., Wick, W., Schlemmer, H. P., & Maier-Hein, K. (2017). Brain tumor segmentation using large receptive field deep convolutional neural networks. In Bildverarbeitung für die Medizin 2017 (pp. 86-91). Springer Vieweg, Berlin, Heidelberg.

Isensee, F., & Maier-Hein, K. H. (2019). An attempt at beating the 3D U-Net. arXiv preprint arXiv:1908.02182.

Isensee, F., Schell, M., Pflueger, I., Brugnara, G., Bonekamp, D., Neuberger, U., ... & Bendszus, M. (2019). Automated brain extraction of multisequence MRI using artificial neural networks. Human brain mapping, 40(17), 4952-4964.

Heller, N., **Isensee, F.**, Maier-Hein, K. H., Hou, X., Xie, C., Li, F., ... & Yao, G. (2019). The state of the art in kidney and kidney tumor segmentation in contrast-enhanced CT imaging: Results of the KiTS19 Challenge. arXiv preprint arXiv:1912.01054.

Brugnara, G., **Isensee, F.**, Neuberger, U., Bonekamp, D., Petersen, J., Diem, R., ... & Maier-Hein, K. (2020). Automated volumetric assessment with artificial neural networks might enable a more accurate assessment of disease burden in patients with multiple sclerosis. European Radiology, 1-9.

Isensee, F., & Maier-Hein, K. H. (2020). OR-UNet: an Optimized Robust Residual U-Net for Instrument Segmentation in Endoscopic Images. arXiv preprint arXiv:2004.12668.

*equal contribution

Appendices

A. nnU-Net details

A.1. Details on nnU-Net's Data Augmentation

The text contained in this section is reproduced from [23]. It was written by me and describes my implementation of data augmentation in nnU-Net.

"A variety of data augmentation techniques is applied during training. All augmentations are computed on the fly on the CPU using background workers. The data augmentation pipeline is implemented with the publicly available *batchgenerators* framework [1]. nnU-Net does not vary the parameters of the data augmentation pipeline between datasets. Sampled patches are initially larger than the patch size used for training. This results in less out of boundary values (here 0) being introduced during data augmentation when rotation and scaling is applied. As a part of the rotation and scaling augmentation, patches are center-cropped to the final target patch size. To ensure that the borders of original images appear in the final patches, preliminary crops may initially extend outside the boundary of the image. Spatial augmentations (rotation, scaling, low resolution simulation) are applied in 3D for the 3D U-Nets and applied in 2D when training the 2D U-Net or a 3D U-Net with anisotropic patch size. A patch size is considered anisotropic if the largest edge length of the patch size is at least three times larger than the smallest. To increase the variability in generated patches, most augmentations are varied with parameters drawn randomly from predefined ranges. In this context, $x \sim U(a, b)$ indicates that x was drawn from a uniform distribution between a and b. Furthermore, all augmentations are applied stochastically according to a predefined probability. The following augmentations are applied by nnU-Net (in the given order):

[1] https://github.com/MIC-DKFZ/batchgenerators

1. **Rotation and Scaling.** Scaling and rotation are applied together for improved speed of computation. This approach reduces the amount of required data interpolations to one. Scaling and rotation are applied with a probability of 0.2 each (resulting in probabilities of 0.16 for only scaling, 0.16 for only rotation and 0.08 for both being triggered). If processing isotropic 3D patches, the angles of rotation (in degrees) α_x, α_y and α_z are each drawn from $U(-30, 30)$. If a patch is anisotropic or 2D, the angle of rotation is sampled from $U(-180, 180)$. If the 2D patch size is anisotropic, the angle is sampled from $U(-15, 15)$. Scaling is implemented via multiplying coordinates with a scaling factor in the voxel grid. Thus, scale factors smaller than one result in a "zoom out" effect while values larger one result in a "zoom in" effect. The scaling factor is sampled from $U(0.7, 1.4)$ for all patch types.
2. **Gaussian Noise.** Zero centered additive Gaussian noise is added to each voxel in the sample independently. This augmentation is applied with a probability of 0.15. The variance of the noise is drawn from $U(0, 0.1)$ (note that the voxel intensities in all samples are close to zero mean and unit variance due to intensity normalization).
3. **Gaussian Blur.** Blurring is applied with a probability of 0.2 per sample. If this augmentation is triggered in a sample, blurring is applied with a probability of 0.5 for each of the associated modalities (resulting in a combined probability of only 0.1 for samples with a single modality). The width (in voxels) of the Gaussian kernel σ is sampled from $U(0.5, 1.5)$ independently for each modality.
4. **Brightness.** Voxel intensities are multiplied by $x \sim U(0.7, 1.3)$ with a probability of 0.15.
5. **Contrast.** Voxel intensities are multiplied by $x \sim U(0.65, 1.5)$ with a probability of 0.15. Following multiplication, the values are clipped to their original value range.
6. **Simulation of low resolution.** This augmentation is applied with a probability of 0.25 per sample and 0.5 per associated modality. Triggered modalities are downsampled by a factor of $U(1, 2)$ using nearest neighbor interpolation and then sampled back up to their original size with cubic interpolation. For 2D patches or anisotropic 3D patches, this augmentation is applied only in 2D leaving the out of plane axis (if applicable) in its original state.
7. **Gamma augmentation.** This augmentation is applied with a probability of 0.15. The patch intensities are scaled to a factor of $[0, 1]$ of their respective value range. Then, a nonlinear intensity transformation is applied per voxel: $i_{new} = i_{old}^{\gamma}$ with $\gamma \sim U(0.7, 1.5)$. The voxel intensities are subsequently scaled back to their original value range. With a probability of 0.15, this augmentation is applied with the voxel intensities being inverted prior to transformation: $(1 - i_{new}) = (1 - i_{old})^{\gamma}$.
8. **Mirroring.** All patches are mirrored with a probability of 0.5 along all axes.

For the full resolution U-Net of the U-net cascade, nnU-Net additionally applies the following augmentations to the segmentation masks generated by the low resolution

3D U-net. Note that the segmentations are stored as one hot encoding.

1. **Binary Operators.** With probability 0.4, a binary operator is applied to all labels in the predicted masks. This operator is randomly chosen from [dilation, erosion, opening, closing]. The structure element is a sphere with radius $r \sim U(1, 8)$. The operator is applied to the labels in random order. Hereby, the one hot encoding property is retained. Dilation of one label, for example, will result in removal of all other labels in the dilated area.
2. **Removal of Connected Components.** With probability 0.2, connected components that are smaller than 15% of the patch size are removed from the one hot encoding.

" [23]

A.2. Summary of nnU-Net Challenge Participations

The text contained in this section is reproduced from [23]. It was written by me and describes my experimental evaluation of nnU-Net on 19 different datasets in the biomedical domain.

"In this section we provide details of all challenge participations. In some challenges, manual intervention regarding the format of input data or the cross-validation data splits was required for compatibility with nnU-Net. For each dataset, we disclose all manual interventions in this section. The most common cause for manual intervention was training cases that were related to each other (such as multiple time points of the same patient) and thus required to be separated for mutual exclusivity between data splits. A detailed description of how to perform this intervention is further provided along with the source code. For each dataset, we run all applicable nnU-Net configurations (2D, 3D fullres, 3D lowres, 3D cascade) in 5-fold cross-validation. All models are trained from scratch without pretraining and trained only on the provided training data of the challenge without external training data. Note that other participants may be using external data in some competitions. For each dataset, nnU-Net subsequently identifies the ideal configuration(s) based on cross-validation and ensembling. Finally, The best configuration is used to predict the test cases. The pipeline generated by nnU-Net is provided for each dataset in the compact representation described in Section A.2.2. We furthermore provide a table containing detailed cross-validation as well as test set results. All leaderboards were last accessed on December 12th, 2019." [23]

A.2.1. Challenge Inclusion Criteria

"When selecting challenges for participation, our goal was to apply nnU-Net to as many different datasets as possible to demonstrate its robustness and flexibility. We applied the following criteria to ensure a rigorous and sound testing environment:

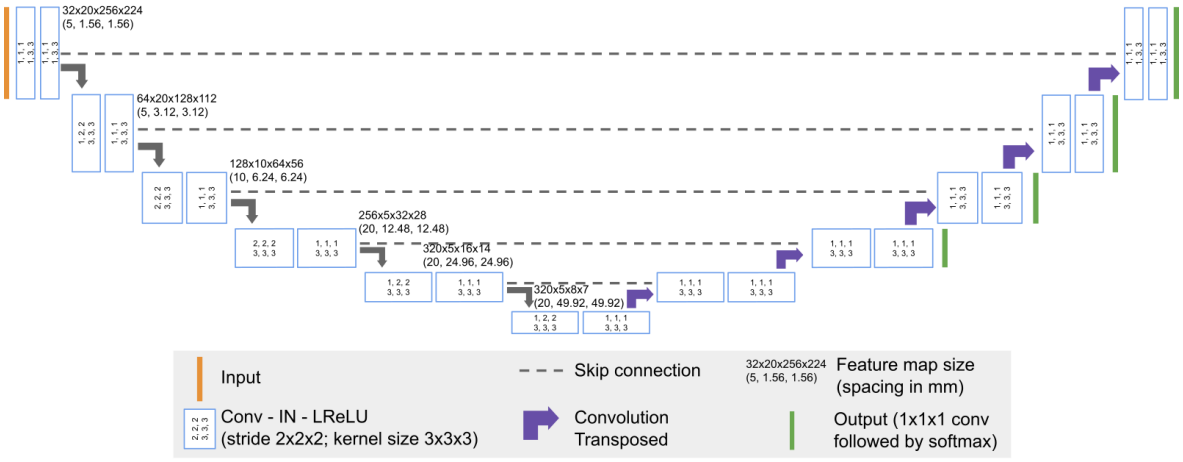

Figure A.1.: "**Decoding the architecture.** We provide all generated architectures in a compact representation from which they can be fully reconstructed if desired. The architecture displayed here can be represented by means of kernel sizes [[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]] and strides [[1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]] (see description in the text)" [23]. Figure reproduced from [23].

1. The task of the challenge is semantic segmentation in any 3D imaging modality with images of any size.
2. Training cases are provided to the challenge participants.
3. Test cases are separate, with the ground truth not being available to the challenge participants.
4. Comparison to results from other participants is possible (e.g. through standardized evaluation with an online platform and a public leaderboard).

The competitions outlined below are the ones who qualified under these criteria and were thus selected for evaluation of nnU-Net. To our knowledge, CREMI [2] is the only competition from the biological domain that meets these criteria." [23]

A.2.2. Compact Architecture Representation

"In the following sections, network architectures generated by nnU-Net will be presented in a compact representation consisting of two lists: one for the convolutional kernel sizes and one for the downsampling strides. As we describe in this section, this representation can be used to fully reconstruct the entire network architecture. The condensed representation is chosen to prevent an excessive amount of figures. Figure A.2.2 exemplary shows the 3D full resolution U-Net for the ACDC dataset (D13). The

[2] https://cremi.org/leaderboard/

architecture has 6 resolution stages. Each resolution stage in both encoder and decoder consists of two computational blocks. Each block is a sequence of (conv - instance norm - leaky ReLU), as described in 4.2. In this figure, one such block is represented by an outlined blue box. Within each box, the stride of the convolution is indicated by the first three numbers (1,1,1 for the uppermost left box) and the kernel size of the convolution is indicated by the second set of numbers (1,3,3 for the uppermost left box). Using this information, along with the template with which our architectures are designed, we can fully describe the presented architecture with the following lists:

- **Convolutional Kernel Sizes**: The kernel sizes of this architecture are [[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]. Note that this list contains 6 elements, matching the 6 resolutions encountered in the encoder. Each element in this list gives the kernel size of the convolutional layers at this resolution (here this is three digits due to the convolutions being three dimensional). Within one resolution, both blocks use the same kernel size. The convolutions in the decoder mirror the encoder (dropping the last entry in the list due to the bottleneck).
- **Downsampling strides**: The strides for downsampling here are [[1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]]. Each downsampling step in the encoder is represented by one entry. A stride of 2 results in a downsampling of factor 2 along that axis which a stride of 1 leaves the size unchanged. Note how the stride initially is [1, 2, 2] due to the spacing discrepancy. This changes the initial spacing of $5 \times 1.56 \times 1.56$ mm to a spacing of $5 \times 3.12 \times 3.12$ mm in the second resolution step. The downsampling strides only apply to the first convolution of each resolution stage in the encoder. The second convolution always has a stride of [1, 1, 1]. Again, the decoder mirrors the encoder, but the stride is used as output stride of the convolution transposed (resulting in appropriate upscaling of feature maps). Outputs of all convolutions transposed have the same shape as the skip connection originating from the encoder.

Segmentation outputs for auxiliary losses are added to all but the two lowest resolution steps." [23]

A.2.3. Medical Segmentation Decathlon

Challenge summary

"The Medical Segmentation Decathlon[3] [29] is a competition that spans 10 different segmentation tasks. These tasks are selected to cover a large proportion of the dataset variability in the medical domain. The overarching goal of the competition was to encourage researchers to develop algorithms that can work with these datasets out of the box without manual intervention. Each of the tasks comes with respective training and

[3] http://medicaldecathlon.com/

test data. A detailed description of datasets can be found on the challenge homepage. Originally, the challenge was divided into two phases: In phase I, 7 datasets were provided to the participants for algorithm development. In phase II, the algorithms were applied to three additional and previously unseen datasets without further changes. Challenge evaluation was performed for the two phases individually and winners were determined based on their performance on the test cases." [23]

Initial version of nnU-Net

"A preliminary version of nnU-Net was developed as part of our entry in this competition, where it achieved the first rank in both phases (see http://medicaldecathlon.com/results.html). We subsequently made the respective challenge report available on arXiv [30].

nnU-Net has since been refined using all ten tasks of the Medical Segmentation Decathlon. The current version of nnU-Net as presented in this publication was again submitted to the open leaderboard (https://decathlon-10.grand-challenge.org/evaluation/results/), and achieved the first rank outperforming the initial nnU-Net as well as other methods that held the state of the art since the original competition [117]." [23]

Application of nnU-Net to the Medical Segmentation Decathlon

"nnU-Net was applied to all ten tasks of the Medical Segmentation Decathlon without any manual intervention." [23]

BrainTumour (D1)

"**Normalization:** Each image is normalized independently by subtracting its mean and dividing by its standard deviation." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 1 x 1	1 x 1 x 1	-
Median image shape at target spacing:	NA x 169 x 138	138 169 138	-
Patch size:	192 x 160	128 x 128 x 128	-
Batch size:	107	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table A.1.: "**Network configurations generated by nnU-Net for the BrainTumour dataset from the Medical Segmentation Decathlon (D1).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	edema	non-enhancing tumor	enhancing tumour	mean
2D	0.7957	0.5985	0.7825	0.7256
3D_fullres *	0.8101	0.6199	0.7934	0.7411
Best Ensemble	0.8106	0.6179	0.7926	0.7404
Postprocessed	0.8101	0.6199	0.7934	0.7411
Test set	0.68	0.47	0.68	0.61

Table A.2.: "**Decathlon BrainTumour (D1) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set") Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

Heart (D2)

"**Normalization:** Each image is normalized independently by subtracting its mean and dividing by its standard deviation." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 1.25 x 1.25	1.37 x 1.25 x 1.25	-
Median image shape at target spacing:	NA x 320 x 232	115 x 320 x 232	-
Patch size:	320 x 256	80 x 192 x 160	-
Batch size:	40	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 1]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3],]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table A.3.: "**Network configurations generated by nnU-Net for the Heart dataset from the Medical Segmentation Decathlon (D2).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	left atrium	mean
2D	0.9090	0.9090
3D_fullres *	0.9328	0.9328
Best Ensemble	0.9268	0.9268
Postprocessed	0.9329	0.9329
Test set	0.93	0.93

Table A.4.: "**Decathlon Heart (D2) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

Liver (D3)

"**Normalization:** Clip to $[-17, 201]$, then subtract 99.40 and finally divide by 39.36." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.7676 x 0.7676	1 x 0.7676 x 0.7676	2.47 x 1.90 x 1.90
Median image shape at target spacing:	NA x 512 x 512	482 x 512 x 512	195 x 207 x 207
Patch size:	512 x 512	128 x 128 x 128	128 x 128 x 128
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table A.5.: "**Network configurations generated by nnU-Net for the Liver dataset from the Medical Segmentation Decathlon (D3).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	liver	cancer	mean
2D	0.9547	0.5637	0.7592
3D_fullres	0.9571	0.6372	0.7971
3D_lowres	0.9563	0.6028	0.7796
3D cascade	0.9600	0.6386	0.7993
Best Ensemble*	0.9613	0.6564	0.8088
Postprocessed	0.9621	0.6600	0.8111
Test set	0.96	0.76	0.86

Table A.6.: "**Decathlon Liver (D3) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D low resolution U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

Hippocampus (D4)

"**Normalization:** Each image is normalized independently by subtracting its mean and dividing by its standard deviation." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 1 x 1	1 x 1 x 1	-
Median image shape at target spacing:	NA x 50 x 35	36 x 50 x 35	-
Patch size:	56 x 40	40 x 56 x 40	-
Batch size:	366	9	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table A.7.: "**Network configurations generated by nnU-Net for the Hippocampus dataset from the Medical Segmentation Decathlon (D4).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	Anterior	Posterior	mean
2D	0.8787	0.8595	0.8691
3D_fullres *	0.8975	0.8807	0.8891
Best Ensemble	0.8962	0.8790	0.8876
Postprocessed	0.8975	0.8807	0.8891
Test set	0.90	0.89	0.895

Table A.8.: "**Decathlon Hippocampus (D4) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

Prostate (D5)

"**Normalization:** Each image is normalized independently by subtracting its mean and dividing by its standard deviation." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.62 x 0.62	3.6 x 0.62 x 0.62	-
Median image shape at target spacing:	NA x 320 x 319	20 x 320 x 319	-
Patch size:	320 x 320	20 x 320 x 256	-
Batch size:	32	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table A.9.: "**Network configurations generated by nnU-Net for the Prostate dataset from the Medical Segmentation Decathlon (D5).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	PZ	TZ	mean
2D	0.6285	0.8380	0.7333
3D_fullres	0.6663	0.8410	0.7537
Best Ensemble *	0.6611	0.8575	0.7593
Postprocessed	0.6611	0.8577	0.7594
Test set	0.77	0.90	0.835

Table A.10.: "**Decathlon Prostate (D5) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

Lung (D6)

"**Normalization:** Clip to $[-1024, 325]$, then subtract -158.58 and finally divide by 324.70." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.79 x 0.79	1.24 x 0.79 x 0.79	2.35 x 1.48 x 1.48
Median image shape at target spacing:	NA x 512 x 512	252 x 512 x 512	133 x 271 x 271
Patch size:	512 x 512	80 x 192 x 160	80 x 192 x 160
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table A.11.: "**Network configurations generated by nnU-Net for the Lung dataset from the Medical Segmentation Decathlon (D6).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	cancer	mean
2D	0.4989	0.4989
3D_fullres	0.7211	0.7211
3D_lowres	0.7109	0.7109
3D cascade	0.6980	0.6980
Best Ensemble*	0.7241	0.7241
Postprocessed	0.7241	0.7241
Test set	0.74	0.74

Table A.12.: "**Decathlon Lung (D6) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D low resolution U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

Pancreas (D7)

"**Normalization:** Clip to $[-96.0, 215.0]$, then subtract 77.99 and finally divide by 75.40." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.8 x 0.8	2.5 x 0.8 x 0.8	2.58 x 1.29 x 1.29
Median image shape at target spacing:	NA x 512 x 512	96 x 512 x 512	93 x 318 x 318
Patch size:	512 x 512	40 x 224 x 224	64 x 192 x 192
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table A.13.: "**Network configurations generated by nnU-Net for the Pancreas dataset from the Medical Segmentation Decathlon (D7).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	pancreas	cancer	mean
2D	0.7738	0.3501	0.5619
3D_fullres	0.8217	0.5274	0.6745
3D_lowres	0.8118	0.5286	0.6702
3D cascade	0.8101	0.5380	0.6741
Best Ensemble *	0.8214	0.5428	0.6821
Postprocessed	0.8214	0.5428	0.6821
Test set	0.82	0.53	0.675

Table A.14.: "**Decathlon Pancreas (D7) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D full resolution U-Net and the 3D U-Net cascade" [23]. Table reproduced from [23].

Hepatic Vessel (D8)

"**Normalization:** Clip to $[-3, 243]$, then subtract 104.37 and finally divide by 52.62." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.8 x 0.8	1.5 x 0.8 x 0.8	2.42 x 1.29 x 1.29
Median image shape at target spacing:	NA x 512 x 512	150 x 512 x 512	93 x 318 x 318
Patch size:	512 x 512	64 x 192 x 192	64 x 192 x 192
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table A.15.: "**Network configurations generated by nnU-Net for the HepaticVessel dataset from the Medical Segmentation Decathlon (D8).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	Vessel	Tumour	mean
2D	0.6180	0.6359	0.6269
3D_fullres	0.6456	0.7217	0.6837
3D_lowres	0.6294	0.7079	0.6687
3D cascade	0.6424	0.7138	0.6781
Best Ensemble *	0.6485	0.7250	0.6867
Postprocessed	0.6485	0.7250	0.6867
Test set	0.66	0.72	0.69

Table A.16.: "**Decathlon HepaticVessel (D8) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D full resolution U-Net and the 3D low resolution U-Net" [23]. Table reproduced from [23].

Spleen (D9)

"**Normalization:** Clip to $[-41, 176]$, then subtract 99.29 and finally divide by 39.47." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.79 x 0.79	1.6 x 0.79 x 0.79	2.77 x 1.38 x 1.38
Median image shape at target spacing:	NA x 512 x 512	187 x 512 x 512	108 x 293 x 293
Patch size:	512 x 512	64 x 192 x 160	64 x 192 x 192
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table A.17.: "**Network configurations generated by nnU-Net for the Spleen dataset from the Medical Segmentation Decathlon (D9).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	spleen	mean
2D	0.9492	0.9492
3D_fullres	0.9638	0.9638
3D_lowres	0.9683	0.9683
3D cascade	0.9714	0.9714
Best Ensemble *	0.9723	0.9723
Postprocessed	0.9724	0.9724
Test set	0.97	0.97

Table A.18.: "**Decathlon Spleen (D9) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D U-Net cascade and the 3D full resolution U-Net" [23]. Table reproduced from [23].

Colon (D10)

"**Normalization:** Clip to $[-30.0, 165.82]$, then subtract 62.18 and finally divide by 32.65." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.78 x 0.78	3 x 0.78 x 0.78	3.09 x 1.55 x 1.55
Median image shape at target spacing:	NA x 512 x 512	150 x 512 x 512	146 x 258 x 258
Patch size:	512 x 512	56 x 192 x 160	96 x 160 x 160
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table A.19.: "**Network configurations generated by nnU-Net for the Colon dataset from the Medical Segmentation Decathlon (D10).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	colon cancer primaries	mean
2D	0.2852	0.2852
3D_fullres	0.4553	0.4553
3D_lowres	0.4538	0.4538
3D cascade *	0.4937	0.4937
Best Ensemble	0.4853	0.4853
Postprocessed	0.4937	0.4937
Test set	0.58	0.58

Table A.20.: "**Decathlon Colon (D10) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D U-Net cascade and the 3D full resolution U-Net" [23]. Table reproduced from [23].

A.2.4. Multi Atlas Labeling Beyond the Cranial Vault: Abdomen (D11)

Challenge summary

"The Multi Atlas Labeling Beyond the Cranial Vault - Abdomen Challenge[4] [36] (denoted BCV for brevity) comprises 30 CT images for training and 20 for testing. The segmentation target are thirteen different organs in the abdomen. " [23]

Application of nnU-Net to BCV

"nnU-Net was applied to the BCV challenge without any manual intervention.

Normalization: Clip to $[-958, 327]$, then subtract 82.92 and finally divide by 136.97." [23]

A.2.5. PROMISE12 (D12)

Challenge summary

"The segmentation target of the PROMISE12 challenge [42] is the prostate in T2 MRI images. 50 training cases with prostate annotations are provided for training. There are 30 test cases which need to be segmented by the challenge participants and are subsequently evaluated on an online platform[5]." [23]

[4] `https://www.synapse.org/Synapse:syn3193805/wiki/217752`
[5] https://promise12.grand-challenge.org/

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.76 x 0.76	3 x 0.76 x 0.76	3.18 x 1.60 x 1.60
Median image shape at target spacing:	NA x 512 x 512	148 x 512 x 512	140 x 243 x 243
Patch size:	512 x 512	48 x 192 x 192	80 x 160 x 160
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table A.21.: "**Network configurations generated by nnU-Net for the BCV challenge (D13).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	1	2	3	4	5	6	7	8
2D	0.8860	0.8131	0.8357	0.6406	0.7724	0.9453	0.8405	0.9128
3D_fullres	0.9083	0.8939	0.8675	0.6632	0.7840	0.9557	0.8816	0.9229
3D_lowres	0.9132	0.9045	0.9132	0.6525	0.7810	0.9554	0.8903	0.9209
3D cascade	0.9166	0.9069	0.9137	0.7036	0.7885	0.9587	0.9037	0.9215
Best Ensemble *	0.9135	0.9065	0.8971	0.6955	0.7897	0.9589	0.9026	0.9248
Postprocessed	0.9135	0.9065	0.8971	0.6959	0.7897	0.9590	0.9026	0.9248
Test set	0.9721	0.9182	0.9578	0.7528	0.8411	0.9769	0.9220	0.9290

	9	10	11	12	13	mean
2D	0.8140	0.7046	0.7367	0.6269	0.5909	0.7784
3D_fullres	0.8638	0.7659	0.8176	0.7148	0.7238	0.8279
3D_lowres	0.8571	0.7469	0.8003	0.6688	0.6851	0.8223
3D cascade	0.8621	0.7722	0.8210	0.7205	0.7214	0.8393
Best Ensemble *	0.8673	0.7746	0.8299	0.7218	0.7287	0.8393
Postprocessed	0.8673	0.7746	0.8299	0.7262	0.7290	0.8397
Test set	0.8809	0.8317	0.8515	0.7887	0.7674	0.8762

Table A.22.: "**Multi Atlas Labeling Beyond the Cranial Vault Abdomen (D11) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. Postprocessing was applied to the model marked with *. This model (incl postprocessing) was used for test set predictions. Note that the Dice scores for the test set are computed with the online platform. Best ensemble on this dataset was the combination of the 3D U-Net cascade and the 3D full resolution U-Net" [23]. Table reproduced from [23].

Application of nnU-Net to PROMISE12

"nnU-Net was applied to the PROMISE12 challenge without any manual intervention.

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.61 x 0.61	2.2 x 0.61 x 0.61	-
Median image shape at target spacing:	NA x 327 x 327	39 x 327 x 327	-
Patch size:	384 x 384	28 x 256 x 256	-
Batch size:	22	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table A.23.: "**Network configurations generated by nnU-Net for the PROMISE12 challenge (D12).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	prostate	mean
2D	0.8932	0.8932
3D_fullres	0.8891	0.8891
Best Ensemble *	0.9029	0.9029
Postprocessed	0.9030	0.9030
Test set	0.9194	0.9194

Table A.24.: "**PROMISE12 (D12) results.** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the scores for the test set are computed with the online platform. The evaluation score of our test set submission is 89.6507. The test set Dice score reported in the table was computed from the detailed submission results (Detailed results available here `https://promise12.grand-challenge.org/evaluation/results/89044a85-6c13-49f4-9742-dea65013e971/`). Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

A.2.6. The Automatic Cardiac Diagnosis Challenge (ACDC) (D13)

Challenge summary

"The Automatic Cardiac Diagnosis Challenge [4] (ACDC) comprises 100 training patients and 50 test patients. The target structures are the cavity of the right ventricle, the myocardium of the left ventricle and the cavity of the left ventricle. All images are cine MRI sequences of which the enddiastolic (ED) and endsystolic (ES) time points of the cardiac cycle were to be segmented. With two time instances per patient, the effective number of training/test images is 200/100." [23]

Application of nnU-Net to ACDC

"Since two time instances of the same patient were provided, we manually interfered with the split for the 5-fold cross-validation of our models to ensure mutual exclusivity of patients between folds. A part from that, nnU-Net was applied without manual intervention.

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 1.56 x 1.56	5 x 1.56 x 1.56	-
Median image shape at target spacing:	NA x 237 x 208	18 x 237 x 208	-
Patch size:	256 x 224	20 x 256 x 224	-
Batch size:	58	3	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table A.25.: "**Network configurations generated by nnU-Net for the ACDC challenge (D13).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	RV	MLV	LVC	mean
2D	0.9053	0.8991	0.9433	0.9159
3D_fullres	0.9059	0.9022	0.9458	0.9179
Best Ensemble *	0.9145	0.9059	0.9479	0.9227
Postprocessed	0.9145	0.9059	0.9479	0.9228
Test set	0.9295	0.9183	0.9407	0.9295

Table A.26.: "**ACDC results (D13).** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform. The online platform reports the Dice scores for enddiastolic and endsystolic time points separately. We averaged these values for a more condensed presentation. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

A.2.7. Liver and Liver Tumor Segmentation Challenge (LiTS) (D14)

Challenge summary

"The Liver and Liver Tumor Segmentation challenge [35] provides 131 training CT images with ground truth annotations for the liver and liver tumors. 70 test images are provided without annotations. The predicted segmentation masks of the test cases are evaluated using the LiTS online platform[6]." [23]

Application of nnU-Net to LiTS

"nnU-Net was applied to the LiTS challenge without any manual intervention.

Normalization: Clip to $[-17, 201]$, then subtract 99.40 and finally divide by 39.39." [23]

[6] https://competitions.codalab.org/competitions/17094

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.77 x 0.77	1 x 0.77 x 0.77	2.47 x 1.90 x 1.90
Median image shape at target spacing:	NA x 512 x 512	482 x 512 x 512	195 x 207 x 207
Patch size:	512 x 512	128 x 128 x 128	128 x 128 x 128
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table A.27.: "**Network configurations generated by nnU-Net for the LiTS challenge (D14).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	liver	cancer	mean
2D	0.9547	0.5603	0.7575
3D_fullres	0.9576	0.6253	0.7914
3D_lowres	0.9585	0.6161	0.7873
3D cascade	0.9609	0.6294	0.7951
Best Ensemble*	0.9618	0.6539	0.8078
Postprocessed	0.9631	0.6543	0.8087
Test set	0.9670	0.7630	0.8650

Table A.28.: "**LiTS results (D14).** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform. Best ensemble on this dataset was the combination of the 3D low resolution U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

A.2.8. Longitudinal multiple sclerosis lesion segmentation challenge (MSLesion) (D15)

Challenge summary

"The longitudinal multiple sclerosis lesion segmentation challenge [39] provides 5 training patients. For each patient, 4 to 5 images acquired at different time points are provided (4 patients with 4 time points each and one patient with 5 time points for a total of 21 images). Each time point is annotated by two different experts, resulting in 42 training annotations (on 21 images). The test set contains 14 patients, again with several time points each, for a total of 61 MRI acquisitions. Test set predictions are

evaluated using the online platform[7]. Each train and test image consists of four MRI modalities: MPRAGE, FLAIR, Proton Density, T2." [23]

Application of nnU-Net to MSLesion

"We manually interfere with the splits in the cross-validation to ensure mutual exclusivity of patients between folds. Each image was annotated by two different experts. We treat these annotations as separate training images (of the same patient), resulting in a training set size of $2 \times 21 = 42$. We do not use the longitudinal nature of the scans and treat each image individually during training and inference.

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 1 x 1	1 x 1 x 1	-
Median image shape at target spacing:	NA x 180 x 137	137 x 180 x 137	-
Patch size:	192 x 160	112 x 128 x 96	-
Batch size:	107	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 1], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table A.29.: "**Network configurations generated by nnU-Net for the MSLesion challenge (D15).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

[7] https://smart-stats-tools.org/lesion-challenge

	lesion	mean
2D	0.7339	0.7339
3D_fullres *	0.7531	0.7531
Best Ensemble	0.7494	0.7494
Postprocessed	0.7531	0.7531
Test set	0.6785	0.6785

Table A.30.: "**MSLesion results (D15).** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform based on the detailed results (which are available here `https://smart-stats-tools.org/sites/lesion_challenge/temp/top25/nnUNetV2_12032019_0903.csv`). The ranking is based on a score, which includes other metrics as well (see [39] for details). The score of our submission is 92.874. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

A.2.9. Combined Healthy Abdominal Organ Segmentation (CHAOS) (D16)

Challenge summary

"The CHAOS challenge [186] is divided into five tasks. Here we focused on Tasks 3 (MRI Liver segmentation) and Task 5 (MRI multiorgan segmentation). Tasks 1, 2 and 4 also included the use of CT images, a modality for which plenty of public data is available (see e.g. BCV and LiTS challenge). To isolate the algorithmic performance of nnU-Net relative to other participants we decided to only use the tasks for which a contamination with external data was unlikely. The target structures of Task 5 are the liver, the spleen and the left and right kidneys. The CHAOS challenge provides 20 training cases. For each training case, there is a T2 images with a corresponding ground truth annotation as well as a T1 acquisition with its own, separate ground truth annotation. The T1 acquisition has two modalities which are co-registered: T1 in-phase and T1 out-phase. Task 3 is a subset of Task 5 with only the liver being the segmentation target. The 20 test cases are evaluated using the online platform[8]." [23]

Application of nnU-Net to CHAOS

"nnU-Net only supports images with a constant number of input modalities. The training cases in CHAOS have either one (T2) or two (T1 in & out phase) modalities. To ensure compatibility with nnU-Net we could have either duplicated the T2 image

[8] https://chaos.grand-challenge.org/

and trained with two input modalities or use only one input modality and treat T1 in phase and out phase as separate training examples. We opted for the latter because this variant results in more (albeit highly correlated) training images. With 20 training patients being provided, this approach resulted in 60 training images. For the cross-validation we ensure that the split is being done on patient level. During inference, nnU-Net will generate two separate predictions for T1 in and out phase which need to be consolidated for test set evaluation. We achieve this by simply averaging the softmax probabilities between the two to generate the final segmentation. We train nnU-Net only for Task 5. Because task 3 represents a subset of Task 5, we extract the liver from our Task 5 predictions and submit it to Task 3.

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 1.66 x 1.66	5.95 x 1.66 x 1.66	-
Median image shape at target spacing:	NA x 195 x 262	45 x 195 x 262	-
Patch size:	224 x 320	40 x 192 x 256	-
Batch size:	45	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [1, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 1, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table A.31.: "**Network configurations generated by nnU-Net for the CHAOS challenge (D16).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	liver	right kidney	left kidney	spleen	mean
2D	0.9132	0.8991	0.8897	0.8720	0.8935
3D_fullres	0.9202	0.9274	0.9209	0.8938	0.9156
Best Ensemble *	0.9184	0.9283	0.9255	0.8911	0.9158
Postprocessed	0.9345	0.9289	0.9212	0.894	0.9197
Test set	-	-	-	-	-

Table A.32.: "**CHAOS results (D16).** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. Postprocessing was applied to the model marked with *. This model (incl postprocessing) was used for test set predictions. Note that the evaluation of the test set was performed with the online platform of the challenge which does not report Dice scores for the individual organs. The score of our submission was 72.44 for Task 5 and 75.10 for Task3 (see [186] for details). Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net" [23]. Table reproduced from [23].

A.2.10. Kidney and Kidney Tumor Segmentation (KiTS) (D17)

Challenge summary

"The Kidney and Kidney Tumor Segmentation challenge [15] was the largest competition (in terms of number of participants) at MICCAI 2019. The target structures are the kidneys and kidney tumors. 210 training and 90 test cases are provided by the challenge organizers. The organizers provide the data both in their original geometry (with voxel spacing varying between cases) as well as interpolated to a common voxel spacing. Evaluation of the test set predictions is done on the online platform[9]. We participated in the original KiTS 2019 MICCAI challenge with a manually designed residual 3D U-Net. This algorithm, described in [165] obtained the first rank in the challenge. For this submission, we did slight modifications to the original training data: Cases 15 and 37 were confirmed to be faulty by the challenge organizers (https://github.com/neheller/kits19/issues/21) which is why we replaced their respective segmentation masks with predictions of one of our networks. We furthermore excluded cases 23, 68, 125 and 133 because we suspected labeling errors in these cases as well. At the time of conducting the experiments for this publication, no revised segmentation masks were provided by the challenge organizers, which is why we reused the modified training dataset for training nnU-Net. After the challenge event at MICCAI 2019, an open leaderboard was created. The original challenge leaderboard is retained at http://results.kits-challenge.org/miccai2019/. All submissions of the original KiTS challenge were mirrored to the open leaderboard. The submission of nnU-Net

[9] https://kits19.grand-challenge.org/

as performed in the context of this manuscript is done on the open leaderboard, where many more competitors have entered since the challenge. As presented in Figure 4.4, nnU-Net sets a new state of the art on the open leaderboard, thus also outperforming our initial, manually optimized solution." [23]

Application of nnU-Net to KiTS

"Since nnU-Net is designed to automatically deal with varying voxel spacings within a dataset, we chose the original, non-interpolated image data as provided by the organizers and let nnU-Net deal with the homogenization of voxel spacing. nnU-Net was applied to the KiTS challenge without any manual intervention.

Normalization: Clip to $[-79, 304]$, then subtract 100.93 and finally divide by 76.90." [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.78 x 0.78	0.78 x 0.78 x 0.78	1.99 x 1.99 x 1.99
Median image shape at target spacing:	NA x 512 x 512	525 x 512 x 512	206 x 201 x 201
Patch size:	512 x 512	128 x 128 x 128	128 x 128 x 128
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table A.33.: "**Network configurations generated by nnU-Net for the KiTS challenge (D17).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	Kidney	Tumor	mean
2D	0.9613	0.7563	0.8588
3D_fullres	0.9702	0.8367	0.9035
3D_lowres	0.9629	0.8420	0.9025
3D cascade	0.9702	0.8546	0.9124
Best Ensemble*	0.9707	0.8620	0.9163
Postprocessed	0.9707	0.8620	0.9163
Test set	-	0.8542	-

Table A.34.: "**KiTS results (D17).** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. Postprocessing was applied to the model marked with *. This model (incl postprocessing) was used for test set predictions. Note that the Dice scores for the test set are computed with the online platform which computes the kidney Dice score based of the union of the kidney and tumor labels whereas nnU-Net always evaluates labels independently, resulting in a missing value for kindey in the table. The reported kindey Dice by the platform (which is not comparable with the value computed by nnU-Net) is 0.9793. Best ensemble on this dataset was the combination of the 3D U-Net cascade and the 3D full resolution U-Net" [23]. Table reproduced from [23].

A.2.11. Segmentation of THoracic Organs at Risk in CT images (SegTHOR) (D18)

Challenge summary

"In the Segmentation of THoracic Organs at Risk in CT images [185] challenge, four abdominal organs (the heart, the aorta, the trachea and the esopahgus) are to be segmented in CT images. 40 training images are provided for training and another 20 images are provided for testing. Evaluation of the test images is done using the online platform[10]." [23]

Application of nnU-Net to SegTHOR

"nnU-Net was applied to the SegTHOR challenge without any manual intervention.

Normalization: Clip to $[-986, 271]$, then subtract 20.78 and finally divide by 180.50." [23]

[10] https://competitions.codalab.org/competitions/21145

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.89 x 0.89	2.50 x 0.89 x 0.89	3.51 x 1.76 x 1.76
Median image shape at target spacing:	NA x 512 x 512	171 x 512 x 512	122 x 285 x 285
Patch size:	512 x 512	64 x 192 x 160	80 x 192 x 160
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table A.35.: "**Network configurations generated by nnU-Net for the SegTHOR challenge (D18).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2" [23]. Table reproduced from [23].

	esophagus	heart	trachea	aorta	mean
2D	0.8181	0.9407	0.9077	0.9277	0.8986
3D_fullres	0.8495	0.9527	0.9055	0.9426	0.9126
3D_lowres	0.8110	0.9464	0.8930	0.9284	0.8947
3D cascade	0.8553	0.9520	0.9045	0.9403	0.9130
Best Ensemble*	0.8545	0.9532	0.9066	0.9427	0.9143
Postprocessed	0.8545	0.9532	0.9083	0.9438	0.9150
Test set	0.8890	0.9570	0.9228	0.9510	0.9300

Table A.36.: "**SegTHOR results (D18).** Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. Postprocessing was applied to the model marked with *. This model (incl postprocessing) was used for test set predictions. Note that the Dice scores for the test set are computed with the online platform. Best ensemble on this dataset was the combination of the 3D U-Net cascade and the 3D full resolution U-Net" [23]. Table reproduced from [23].

A.2.12. Challenge on Circuit Reconstruction from Electron Microscopy Images (CREMI) (D19)

Challenge summary

"The Challenge on Circuit Reconstruction from Electron Microscopy Images is subdivided into three tasks. The synaptic cleft segmentation task can be formulated as semantic segmentation (as opposed to e.g. instance segmentation) and is thus compatible with nnU-Net. In this task, the segmentation target is the cell membrane in locations where the cells are forming a synapse. The dataset consists of serial section Transmission Electron Microscopy scans of the *Drosophila melanogaster* brain. Three

volumes are provided for training and another three are provided for testing. Test set evaluation is done using the online platform[11].” [23]

Application of nnU-Net to CREMI

”Since to the number of training images is lower than the number of splits, we cannot run a 5-fold cross-validation. Thus, we trained 5 model instances, each of them on all three training volumes and subsequently ensembled these models for test set prediction. Because this training scheme leaves no validation data, selection of the best of three model configurations as performed by nnU-Net after cross-validation was not possible. Hence, we intervened by only configuring and training the 3D full resolution configuration.

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation.” [23]

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	-	40 x 4 x 4	-
Median image shape at target spacing:	-	125 x 1250 x 1250	-
Patch size:	-	24 x 256 x256	-
Batch size:	-	2	-
Downsampling strides:	-	[[1, 2, 2], [1, 2, 2], [1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	-	[[1, 3, 3], [1, 3, 3], [1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table A.37.: ”**Network configurations generated by nnU-Net for the CREMI challenge (D19).** For more information on how to decode downsampling strides and kernel sizes into an architecture, see A.2.2” [23]. Table reproduced from [23].

Results

”Because our training scheme for this challenge left no validation data, a performance estimate as given for the other datastes is not available for CREMI. The CREMI test set is evaluated by the online platform. The evaluation metric is the so called CREMI score, a description of which is available here `https://cremi.org/metrics/`. Dice scores for the test set are not reported. The CREMI score of our test set submission was 74.96 (lower is better).” [23]

[11] https://cremi.org/

Bibliography

[1] S. Nikolov, S. Blackwell, R. Mendes, J. De Fauw, C. Meyer, C. Hughes, H. Askham, B. Romera-Paredes, A. Karthikesalingam, C. Chu *et al.*, "Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy," *arXiv preprint arXiv:1809.04430*, 2018.

[2] B. L. Edlow, A. Mareyam, A. Horn, J. R. Polimeni, T. Witzel, M. D. Tisdall, J. C. Augustinack, J. P. Stockmann, B. R. Diamond, A. Stevens *et al.*, "7 tesla mri of the ex vivo human brain at 100 micron resolution," *Scientific data*, vol. 6, no. 1, pp. 1–10, 2019.

[3] S. Zhang, A. A. Joseph, D. Voit, S. Schaetz, K.-D. Merboldt, C. Unterberg-Buchwald, A. Hennemuth, J. Lotz, and J. Frahm, "Real-time magnetic resonance imaging of cardiac function and flow—recent progress," *Quantitative imaging in medicine and surgery*, vol. 4, no. 5, p. 313, 2014.

[4] O. Bernard, A. Lalande, C. Zotti, F. Cervenansky, X. Yang, P.-A. Heng, I. Cetin, K. Lekadir, O. Camara, M. A. G. Ballester *et al.*, "Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: Is the problem solved?" *IEEE TMI*, vol. 37, no. 11, pp. 2514–2525, 2018.

[5] R. C. of Radiologists., "Clinical radiology uk workforce census 2017 report," 2018. [Online]. Available: https://www.rcr.ac.uk/system/files/publication/field_publication_files/clinical-radiology-uk-workforce-census-report-2018.pdf

[6] E. Sokolovskaya, T. Shinde, R. B. Ruchman, A. J. Kwak, S. Lu, Y. K. Shariff, E. F. Wiggins, and L. Talangbayan, "The effect of faster reporting speed for imaging studies on the number of misses and interpretation errors: a pilot study," *Journal of the American College of Radiology*, vol. 12, no. 7, pp. 683–688, 2015.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] P. Kickingereder, F. Isensee, I. Tursunova, J. Petersen, U. Neuberger, D. Bonekamp, G. Brugnara, M. Schell, T. Kessler, M. Foltyn *et al.*, "Automated quantitative tumour response assessment of mri in neuro-oncology with artificial neural networks: a multicentre, retrospective study," *The Lancet Oncology*, vol. 20, no. 5, pp. 728–740, 2019.

[11] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[12] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin *et al.*, "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nature medicine*, vol. 24, no. 9, pp. 1342–1350, 2018.

[13] S. G. Armato III, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, B. Zhao, D. R. Aberle, C. I. Henschke, E. A. Hoffman *et al.*, "The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans," *Medical physics*, vol. 38, no. 2, pp. 915–931, 2011.

[14] T. Drew, M. L.-H. Võ, and J. M. Wolfe, "The invisible gorilla strikes again: Sustained inattentional blindness in expert observers," *Psychological science*, vol. 24, no. 9, pp. 1848–1853, 2013.

[15] N. Heller, N. Sathianathen, A. Kalapara, E. Walczak, K. Moore, H. Kaluzniak, J. Rosenberg, P. Blake, Z. Rengel, M. Oestreich *et al.*, "The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes," *arXiv preprint arXiv:1904.00445*, 2019.

[16] L. Maier-Hein, M. Eisenmann, A. Reinke, S. Onogur, M. Stankovic, P. Scholz, T. Arbel, H. Bogunovic, A. P. Bradley, A. Carass *et al.*, "Why rankings of biomedical image analysis competitions should be interpreted with care," *Nature communications*, vol. 9, no. 1, p. 5217, 2018.

[17] M. J. van den Bent, J. S. Wefel, D. Schiff, M. J. Taphoorn, K. Jaeckle, L. Junck, T. Armstrong, A. Choucair, A. D. Waldman, T. Gorlia *et al.*, "Response as-

sessment in neuro-oncology (a report of the rano group): assessment of outcome in trials of diffuse low-grade gliomas," *The lancet oncology*, vol. 12, no. 6, pp. 583–593, 2011.

[18] A. Kutikov and R. G. Uzzo, "The renal nephrometry score: a comprehensive standardized system for quantitating renal tumor size, location and depth," *The Journal of urology*, vol. 182, no. 3, pp. 844–853, 2009.

[19] V. Ficarra, G. Novara, S. Secco, V. Macchi, A. Porzionato, R. De Caro, and W. Artibani, "Preoperative aspects and dimensions used for an anatomical (padua) classification of renal tumours in patients who are candidates for nephron-sparing surgery," *European urology*, vol. 56, no. 5, pp. 786–793, 2009.

[20] H. J. Aerts, E. R. Velazquez, R. T. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, J. Bussink, R. Monshouwer, B. Haibe-Kains, D. Rietveld *et al.*, "Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach," *Nature communications*, vol. 5, no. 1, pp. 1–9, 2014.

[21] P. Kickingereder, U. Neuberger, D. Bonekamp, P. L. Piechotta, M. Götz, A. Wick, M. Sill, A. Kratz, R. T. Shinohara, D. T. Jones *et al.*, "Radiomic subtyping improves disease stratification beyond key molecular, clinical, and standard imaging characteristics in patients with glioblastoma," *Neuro-oncology*, vol. 20, no. 6, pp. 848–857, 2018.

[22] P. Kickingereder, M. Götz, J. Muschelli, A. Wick, U. Neuberger, R. T. Shinohara, M. Sill, M. Nowosielski, H.-P. Schlemmer, A. Radbruch *et al.*, "Large-scale radiomic profiling of recurrent glioblastoma identifies an imaging predictor for stratifying anti-angiogenic treatment response," *Clinical Cancer Research*, vol. 22, no. 23, pp. 5765–5771, 2016.

[23] F. Isensee, P. F. Jäger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "Automated design of deep learning methods for biomedical image segmentation," *arXiv preprint arXiv:1904.08128*, 2019.

[24] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications*. Springer, 2016, pp. 179–187.

[25] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 565–571.

[26] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng, "H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from ct volumes," *IEEE TMI*, vol. 37, no. 12, pp. 2663–2674, 2018.

[27] F. Isensee, P. F. Jaeger, P. M. Full, I. Wolf, S. Engelhardt, and K. H. Maier-Hein, "Automatic cardiac disease assessment on cine-mri via time-series segmentation and domain specific features," in *International workshop on statistical atlases and computational models of the heart*. Springer, 2017, pp. 120–129.

[28] N. Heller, F. Isensee, K. H. Maier-Hein, X. Hou, C. Xie, F. Li, Y. Nan, G. Mu, Z. Lin, M. Han *et al.*, "The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: Results of the kits19 challenge," *arXiv preprint arXiv:1912.01054*, 2019.

[29] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithmsdelldatagrowth," *arXiv preprint arXiv:1902.09063*, 2019.

[30] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert *et al.*, "nnu-net: Self-adapting framework for u-net-based medical image segmentation," *arXiv preprint arXiv:1809.10486*, 2018.

[31] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. A. Eslami, D. J. Rezende, and O. Ronneberger, "A probabilistic u-net for segmentation of ambiguous images," in *Advances in Neural Information Processing Systems*, 2018, pp. 6965–6975.

[32] S. A. Kohl, B. Romera-Paredes, K. H. Maier-Hein, D. J. Rezende, S. Eslami, P. Kohli, A. Zisserman, and O. Ronneberger, "A hierarchical probabilistic u-net for modeling multi-scale ambiguities," *arXiv preprint arXiv:1905.13077*, 2019.

[33] C. F. Baumgartner, K. C. Tezcan, K. Chaitanya, A. M. Hötker, U. J. Muehlematter, K. Schawkat, A. S. Becker, O. Donati, and E. Konukoglu, "Phiseg: Capturing uncertainty in medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 119–127.

[34] R. McKinley, R. Meier, and R. Wiest, "Ensembles of densely-connected cnns with label-uncertainty for brain tumor segmentation," in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 456–465.

[35] P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen, Q. Dou, C.-W. Fu, X. Han, P.-A. Heng, J. Hesser *et al.*, "The liver tumor segmentation benchmark (lits)," *arXiv preprint arXiv:1901.04056*, 2019.

[36] B. Landman, Z. Xu, J. Eugenio Igelsias, M. Styner, T. Langerak, and A. Klein,

"Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge," 2015.

[37] H. R. Roth, A. Farag, E. B. Turkbey, L. Lu, J. Liu, and R. M. Summers, "Data from pancreas-ct," *The Cancer Imaing Archive*, 2016.

[38] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.

[39] A. Carass, S. Roy, A. Jog, J. L. Cuzzocreo, E. Magrath, A. Gherman, J. Button, J. Nguyen, F. Prados, C. H. Sudre *et al.*, "Longitudinal multiple sclerosis lesion segmentation: resource and challenge," *NeuroImage*, vol. 148, pp. 77–102, 2017.

[40] O. Maier, B. H. Menze, J. von der Gablentz, L. Häni, M. P. Heinrich, M. Liebrand, S. Winzeck, A. Basit, P. Bentley, L. Chen *et al.*, "Isles 2015-a public evaluation benchmark for ischemic stroke lesion segmentation from multispectral mri," *Medical image analysis*, vol. 35, pp. 250–269, 2017.

[41] A. H. Kadish, D. Bello, J. P. Finn, R. O. Bonow, A. Schaechter, H. Subacius, C. Albert, J. P. Daubert, C. G. Fonseca, and J. J. Goldberger, "Rationale and design for the defibrillators to reduce risk by magnetic resonance imaging evaluation (determine) trial," *Journal of cardiovascular electrophysiology*, vol. 20, no. 9, pp. 982–987, 2009.

[42] G. Litjens, R. Toth, W. van de Ven, C. Hoeks, S. Kerkstra, B. van Ginneken, G. Vincent, G. Guillard, N. Birbeck, J. Zhang *et al.*, "Evaluation of prostate segmentation algorithms for mri: the promise12 challenge," *Med Image Analysis*, vol. 18, no. 2, pp. 359–373, 2014.

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*. Springer, 2015, pp. 234–241.

[44] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2016, pp. 424–432.

[45] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki *et al.*, "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge," *arXiv preprint arXiv:1811.02629*, 2018.

[46] T. Heimann and H.-P. Meinzer, "Statistical shape models for 3d medical image segmentation: a review," *Medical image analysis*, vol. 13, no. 4, pp. 543–563, 2009.

[47] S. Durrleman, X. Pennec, A. Trouvé, J. Braga, G. Gerig, and N. Ayache, "Toward a comprehensive framework for the spatiotemporal statistical analysis of longitudinal shape data," *International journal of computer vision*, vol. 103, no. 1, pp. 22–59, 2013.

[48] G. Gerig, M. Styner, D. Jones, D. Weinberger, and J. Lieberman, "Shape analysis of brain ventricles using spharm," in *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA 2001)*. IEEE, 2001, pp. 171–178.

[49] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Training models of shape from sets of examples," in *BMVC92*. Springer, 1992, pp. 9–18.

[50] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[51] F. De La Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 117–142, 2003.

[52] C. J. Twining and C. J. Taylor, "The use of kernel principal component analysis to model data distributions," *Pattern Recognition*, vol. 36, no. 1, pp. 217–227, 2003.

[53] T. F. Cootes and C. J. Taylor, "Using grey-level models to improve active shape model search," in *Proceedings of 12th international conference on pattern recognition*, vol. 1. IEEE, 1994, pp. 63–67.

[54] G. Behiels, F. Maes, D. Vandermeulen, and P. Suetens, "Evaluation of image features and search strategies for segmentation of bone structures in radiographs using active shape models," *Medical Image Analysis*, vol. 6, no. 1, pp. 47–62, 2002.

[55] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[56] R. Cuingnet, R. Prevost, D. Lesage, L. D. Cohen, B. Mory, and R. Ardon, "Automatic detection and segmentation of kidneys in 3d ct images using random forests," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2012, pp. 66–74.

[57] T. Norajitra and K. H. Maier-Hein, "3d statistical shape models incorporating landmark-wise random regression forests for omni-directional landmark detection," *IEEE transactions on medical imaging*, vol. 36, no. 1, pp. 155–168, 2016.

[58] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annual review of biomedical engineering*, vol. 2, no. 1, pp. 315–337, 2000.

[59] N. T. Doan, J. O. de Xivry, and B. Macq, "Effect of inter-subject variation on the accuracy of atlas-based segmentation applied to human brain structures," in *Medical Imaging 2010: Image Processing*, vol. 7623. International Society for Optics and Photonics, 2010, p. 76231S.

[60] A. Klein, B. Mensh, S. Ghosh, J. Tourville, and J. Hirsch, "Mindboggle: automated brain labeling with multiple atlases," *BMC medical imaging*, vol. 5, no. 1, p. 7, 2005.

[61] R. A. Heckemann, J. V. Hajnal, P. Aljabar, D. Rueckert, and A. Hammers, "Automatic anatomical brain mri segmentation combining label propagation and decision fusion," *NeuroImage*, vol. 33, no. 1, pp. 115–126, 2006.

[62] J. E. Iglesias and M. R. Sabuncu, "Multi-atlas segmentation of biomedical images: a survey," *Medical image analysis*, vol. 24, no. 1, pp. 205–219, 2015.

[63] B. Kim, J. Kim, J.-G. Lee, D. H. Kim, S. H. Park, and J. C. Ye, "Unsupervised deformable image registration using cycle-consistent cnn," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 166–174.

[64] M. P. Heinrich, "Closing the gap between deep and conventional image registration using probabilistic dense displacement networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 50–58.

[65] J. Esteban, M. Grimm, M. Unberath, G. Zahnd, and N. Navab, "Towards fully automatic x-ray to ct registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 631–639.

[66] D. M. Gavrila and V. Philomin, "Real-time object detection for" smart" vehicles," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1. IEEE, 1999, pp. 87–93.

[67] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.

[68] C. Sommer, C. Straehle, U. Koethe, and F. A. Hamprecht, "Ilastik: Interactive learning and segmentation toolkit," in *2011 IEEE international symposium on biomedical imaging: From nano to macro*. IEEE, 2011, pp. 230–233.

[69] S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy *et al.*, "ilastik: Interactive machine learning for (bio) image analysis," *Nature Methods*, pp. 1–7, 2019.

[70] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[71] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[72] J. Kleesiek, A. Biller, G. Urban, U. Kothe, M. Bendszus, and F. Hamprecht, "Ilastik for multi-modal brain tumor segmentation," *Proceedings MICCAI BraTS (Brain Tumor Segmentation Challenge)*, pp. 12–17, 2014.

[73] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1397–1409, 2012.

[74] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[75] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán, "Multiscale conditional random fields for image labeling," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2. IEEE, 2004, pp. II–II.

[76] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[77] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[78] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[79] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[80] M. ul Hassan. (2018) Vgg16 – convolutional network for classification and detection. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/

[81] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[82] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[83] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *NIPS*, 2015, pp. 2962–2970.

[84] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[85] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[86] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[87] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.

[88] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2012.

[89] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[90] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[91] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[92] K. Kamnitsas, E. Ferrante, S. Parisot, C. Ledig, A. V. Nori, A. Criminisi, D. Rueckert, and B. Glocker, "Deepmedic for brain tumor segmentation," in *International workshop on Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries*. Springer, 2016, pp. 138–149.

[93] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 11–19.

[94] M. Khened, V. Alex, and G. Krishnamurthi, "Densely connected fully convolutional network for short-axis cardiac cine mr image segmentation and heart diagnosis using random forest," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2017, pp. 140–151.

[95] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel 'squeeze & excitation'in fully convolutional networks," in *MICCAI*. Springer, 2018, pp. 421–429.

[96] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[97] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[98] K. Kamnitsas, W. Bai, E. Ferrante, S. McDonagh, M. Sinclair, N. Pawlowski, M. Rajchl, M. Lee, B. Kainz, D. Rueckert *et al.*, "Ensembles of multiple models and architectures for robust brain tumour segmentation," in *International MICCAI Brainlesion Workshop*. Springer, 2017, pp. 450–462.

[99] A. Myronenko, "3d mri brain tumor segmentation using autoencoder regularization," in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 311–320.

[100] Q. Yao, M. Wang, Y. Chen, W. Dai, H. Yi-Qi, L. Yu-Feng, T. Wei-Wei, Y. Qiang, and Y. Yang, "Taking human out of learning applications: A survey on automated machine learning," *arXiv preprint arXiv:1810.13306*, 2018.

[101] M.-A. Zöller and M. F. Huber, "Benchmark and survey of automated machine learning frameworks," *arXiv preprint arXiv:1904.12054*, 2019.

[102] T. M. Mitchell *et al.*, "Machine learning. 1997," *Burr Ridge, IL: McGraw Hill*, vol. 45, no. 37, pp. 870–877, 1997.

[103] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 473–480.

[104] H. H. Hoos, "Automated algorithm configuration and parameter tuning," in *Autonomous search*. Springer, 2011, pp. 37–71.

[105] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.

[106] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *arXiv preprint arXiv:1908.00709*, 2019.

[107] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[108] G. Katz, E. C. R. Shin, and D. Song, "Explorekit: Automatic feature generation and selection," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 979–984.

[109] R. Kohavi, G. H. John *et al.*, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[110] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: a unifying framework for information theoretic feature selection," *The journal of machine learning research*, vol. 13, no. 1, pp. 27–66, 2012.

[111] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[112] I. Bello, B. Zoph, V. Vasudevan, and Q. V. Le, "Neural optimizer search with reinforcement learning," *arXiv preprint arXiv:1709.07417*, 2017.

[113] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," *arXiv preprint arXiv:1611.02167*, 2016.

[114] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on artificial intelligence*, vol. 33, 2019, pp. 4780–4789.

[115] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," *arXiv preprint arXiv:1802.03268*, 2018.

[116] S. Kim, I. Kim, S. Lim, W. Baek, C. Kim, H. Cho, B. Yoon, and T. Kim, "Scalable neural architecture search for 3d medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 220–228.

[117] Q. Yu, D. Yang, H. Roth, Y. Bai, Y. Zhang, A. L. Yuille, and D. Xu, "C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation," *arXiv preprint arXiv:1912.09628*, 2019.

[118] T. Sørensen, T. Sørensen, T. Sørensen, T. SORENSEN, T. Sorensen, T. Sorensen, and T. Biering-Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons," 1948.

[119] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[120] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are gans created equal? a large-scale study," in *Advances in neural information processing systems*, 2018, pp. 700–709.

[121] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," *arXiv preprint arXiv:2003.08505*, 2020.

[122] H. Ohgaki and P. Kleihues, "Population-based studies on incidence, survival rates, and genetic alterations in astrocytic and oligodendroglial gliomas," *Journal of Neuropathology & Experimental Neurology*, vol. 64, no. 6, pp. 479–489, 2005.

[123] S. Thust, S. Heiland, A. Falini, H. Jäger, A. Waldman, P. Sundgren, C. Godi, V. Katsaros, A. Ramos, N. Bargallo *et al.*, "Glioma imaging in europe: a survey of 220 centres and recommendations for best clinical practice," *European radiology*, vol. 28, no. 8, pp. 3306–3317, 2018.

[124] P. Y. Wen, S. M. Chang, M. J. Van den Bent, M. A. Vogelbaum, D. R. Macdonald, and E. Q. Lee, "Response assessment in neuro-oncology clinical trials," *Journal of Clinical Oncology*, vol. 35, no. 21, p. 2439, 2017.

[125] W. Wick, T. Gorlia, M. Bendszus, M. Taphoorn, F. Sahm, I. Harting, A. A. Brandes, W. Taal, J. Domont, A. Idbaih *et al.*, "Lomustine and bevacizumab in progressive glioblastoma," *New England Journal of Medicine*, vol. 377, no. 20, pp. 1954–1963, 2017.

[126] R. L. Korn and J. J. Crowley, "Overview: progression-free survival as an endpoint in clinical trials with solid tumors," 2013.

[127] D. Chow, J. Qi, X. Guo, V. Miloushev, F. Iwamoto, J. Bruce, A. Lassman, L. Schwartz, A. Lignelli, B. Zhao *et al.*, "Semiautomated volumetric measurement on postcontrast mr imaging for analysis of recurrent and residual disease in glioblastoma multiforme," *American Journal of Neuroradiology*, vol. 35, no. 3, pp. 498–503, 2014.

[128] A. G. Sorensen, S. Patel, C. Harmath, S. Bridges, J. Synnott, A. Sievers, Y.-H. Yoon, E. J. Lee, M. C. Yang, R. F. Lewis *et al.*, "Comparison of diameter and perimeter methods for tumor volume calculation," *Journal of Clinical Oncology*, vol. 19, no. 2, pp. 551–557, 2001.

[129] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein, "No new-net," in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 234–244.

[130] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein, "Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge," in *International MICCAI Brainlesion Workshop*. Springer, 2017, pp. 287–297.

[131] W. Wick, R. Stupp, T. Gorlia, M. Bendszus, F. Sahm, J. E. Bromberg, A. A. Brandes, M. J. Vos, J. Domont, A. Idbaih *et al.*, "Phase ii part of eortc study 26101: The sequence of bevacizumab and lomustine in patients with first recurrence of a glioblastoma." 2016.

[132] S. M. Smith, "Fast robust automated brain extraction," *Human brain mapping*, vol. 17, no. 3, pp. 143–155, 2002.

[133] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3d active contour segmentation of anatomical structures: significantly improved efficiency and reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.

[134] R. T. Shinohara, E. M. Sweeney, J. Goldsmith, N. Shiee, F. J. Mateen, P. A. Calabresi, S. Jarso, D. L. Pham, D. S. Reich, C. M. Crainiceanu *et al.*, "Statistical normalization techniques for magnetic resonance imaging," *NeuroImage: Clinical*, vol. 6, pp. 9–19, 2014.

[135] N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee, "N4itk: improved n3 bias correction," *IEEE transactions on medical imaging*, vol. 29, no. 6, pp. 1310–1320, 2010.

[136] B. Kayalibay, G. Jensen, and P. van der Smagt, "Cnn-based segmentation of medical imaging data," *arXiv preprint arXiv:1701.03056*, 2017.

[137] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[138] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.

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

[140] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[141] A. Hernández-García and P. König, "Data augmentation instead of explicit regularization," *arXiv preprint arXiv:1806.03852*, 2018.

[142] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," *Scientific data*, vol. 4, p. 170117, 2017.

[143] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, "Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection," *The Cancer Imaging Archive*, 2017.

[144] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, "Segmentation labels and radiomic features for the pre-operative scans of the tcga-gbm collection. the cancer imaging archive," 2017.

[145] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[146] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, "Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks," in *International MICCAI brainlesion workshop*. Springer, 2017, pp. 178–190.

[147] S. Lee, S. Purushwalkam, M. Cogswell, D. Crandall, and D. Batra, "Why m heads are better than one: Training a diverse ensemble of deep networks," *arXiv preprint arXiv:1511.06314*, 2015.

[148] J. Bertels, T. Eelbode, M. Berman, D. Vandermeulen, F. Maes, R. Bisschops, and M. B. Blaschko, "Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 92–100.

[149] H. D. White, R. M. Norris, M. A. Brown, P. W. Brandt, R. Whitlock, and C. J. Wild, "Left ventricular end-systolic volume as the major determinant of survival after recovery from myocardial infarction." *Circulation*, vol. 76, no. 1, pp. 44–51, 1987.

[150] R. M. Norris, H. D. White, D. B. Cross, C. J. Wild, and R. M. Whitlock, "Prognosis after recovery from myocardial infarction: the relative importance of cardiac dilatation and coronary stenoses," *European heart journal*, vol. 13, no. 12, pp. 1611–1618, 1992.

[151] M. H. Hwang, K. Hammermeister, C. Oprian, W. Henderson, G. Bousvaros, M. Wong, D. Miller, E. Folland, and G. Sethi, "Preoperative identification of patients likely to have left ventricular dysfunction after aortic valve replacement. participants in the veterans administration cooperative study on valvular heart disease." *Circulation*, vol. 80, no. 3 Pt 1, pp. I65–76, 1989.

[152] C. A. Miller, P. Jordan, A. Borg, R. Argyle, D. Clark, K. Pearce, and M. Schmitt, "Quantification of left ventricular indices from ssfp cine imaging: Impact of real-world variability in analysis methodology and utility of geometric modeling," *Journal of Magnetic Resonance Imaging*, vol. 37, no. 5, pp. 1213–1222, 2013.

[153] A. F. members, P. M. Elliott, A. Anastasakis, M. A. Borger, M. Borggrefe, F. Cecchi, P. Charron, A. A. Hagege, A. Lafont, G. Limongelli *et al.*, "2014 esc guidelines on diagnosis and management of hypertrophic cardiomyopathy: the task force for the diagnosis and management of hypertrophic cardiomyopathy of the european society of cardiology (esc)," *European heart journal*, vol. 35, no. 39, pp. 2733–2779, 2014.

[154] S. Queirós, D. Barbosa, B. Heyde, P. Morais, J. L. Vilaça, D. Friboulet, O. Bernard, and J. D'hooge, "Fast automatic myocardial segmentation in 4d cine cmr datasets," *Medical image analysis*, vol. 18, no. 7, pp. 1115–1131, 2014.

[155] M. Avendi, A. Kheradvar, and H. Jafarkhani, "A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac mri," *Medical image analysis*, vol. 30, pp. 108–119, 2016.

[156] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[157] C. F. Baumgartner, L. M. Koch, M. Pollefeys, and E. Konukoglu, "An exploration of 2d and 3d deep learning techniques for cardiac mr image segmentation," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2017, pp. 111–119.

[158] C. Zotti, Z. Luo, O. Humbert, A. Lalande, and P.-M. Jodoin, "Gridnet with automatic shape prior registration for automatic mri cardiac segmentation," in

International Workshop on Statistical Atlases and Computational Models of the Heart. Springer, 2017, pp. 73–81.

[159] Y. Jang, Y. Hong, S. Ha, S. Kim, and H.-J. Chang, "Automatic segmentation of lv and rv in cardiac mri," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2017, pp. 161–169.

[160] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum, "Automatic segmentation and disease classification using cardiac cine mr images," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2017, pp. 101–110.

[161] M.-M. Rohé, M. Sermesant, and X. Pennec, "Automatic multi-atlas segmentation of myocardium with svf-net," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2017, pp. 170–177.

[162] J. Patravali, S. Jain, and S. Chilamkurthy, "2d-3d fully convolutional neural networks for cardiac mr segmentation," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2017, pp. 130–139.

[163] E. Grinias and G. Tziritas, "Fast fully-automatic cardiac segmentation in mri using mrf model optimization, substructures tracking and b-spline smoothing," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2017, pp. 91–100.

[164] X. Yang, C. Bian, L. Yu, D. Ni, and P.-A. Heng, "Class-balanced deep neural network for automatic ventricular structure segmentation," in *International workshop on statistical atlases and computational models of the heart*. Springer, 2017, pp. 152–160.

[165] F. Isensee and K. H. Maier-Hein, "An attempt at beating the 3d u-net," *arXiv preprint arXiv:1908.02182*, 2019.

[166] J. M. Hollingsworth, D. C. Miller, S. Daignault, and B. K. Hollenbeck, "Rising incidence of small renal masses: a need to reassess treatment effect," *Journal of the National Cancer Institute*, vol. 98, no. 18, pp. 1331–1334, 2006.

[167] U. Capitanio and F. Montorsi, "Renal cancer," *The Lancet*, vol. 387, no. 10021, pp. 894–906, 2016.

[168] C. J. Robson, "Radical nephrectomy for renal cell carcinoma," *The Journal of urology*, vol. 89, no. 1, pp. 37–42, 1963.

[169] E. Scosyrev, E. M. Messing, R. Sylvester, S. Campbell, and H. Van Poppel, "Renal function after nephron-sparing surgery versus radical nephrectomy: results

from eortc randomized trial 30904," *European urology*, vol. 65, no. 2, pp. 372–377, 2014.

[170] A. C. Uzosike, H. D. Patel, R. Alam, Z. R. Schwen, M. Gupta, M. A. Gorin, M. H. Johnson, H. Gausepohl, M. F. Riffon, B. J. Trock *et al.*, "Growth kinetics of small renal masses on active surveillance: variability and results from the dissrm registry," *The Journal of urology*, vol. 199, no. 3, pp. 641–648, 2018.

[171] P. O. Richard, M. A. Jewett, J. R. Bhatt, A. J. Evans, N. Timilsina, and A. Finelli, "Active surveillance for renal neoplasms with oncocytic features is safe," *The Journal of urology*, vol. 195, no. 3, pp. 581–587, 2016.

[172] I. Millet, F. C. Doyon, D. Hoa, R. Thuret, S. Merigeaud, I. Serre, and P. Taourel, "Characterization of small solid renal lesions: can benign and malignant tumors be differentiated with ct?" *American journal of roentgenology*, vol. 197, no. 4, pp. 887–896, 2011.

[173] M. N. Simmons, S. P. Hillyer, B. H. Lee, A. F. Fergany, J. Kaouk, and S. C. Campbell, "Diameter-axial-polar nephrometry: integration and optimization of renal and centrality index scoring systems," *The Journal of urology*, vol. 188, no. 2, pp. 384–390, 2012.

[174] M. Spaliviero, B. Y. Poon, O. Aras, P. L. Di Paolo, G. B. Guglielmetti, C. Z. Coleman, C. A. Karlo, M. L. Bernstein, D. D. Sjoberg, P. Russo *et al.*, "Interobserver variability of renal, padua, and centrality index nephrometry score systems," *World journal of urology*, vol. 33, no. 6, pp. 853–858, 2015.

[175] A. Kutikov, M. C. Smaldone, B. L. Egleston, B. J. Manley, D. J. Canter, J. Simhan, S. A. Boorjian, R. Viterbo, D. Y. Chen, R. E. Greenberg *et al.*, "Anatomic features of enhancing renal masses predict malignant and high-grade pathology: a preoperative nomogram using the renal nephrometry score," *European urology*, vol. 60, no. 2, pp. 241–248, 2011.

[176] M. H. Hayn, T. Schwaab, W. Underwood, and H. L. Kim, "Renal nephrometry score predicts surgical outcomes of laparoscopic partial nephrectomy," *BJU international*, vol. 108, no. 6, pp. 876–881, 2011.

[177] Z. Okhunov, S. Rais-Bahrami, A. K. George, N. Waingankar, B. Duty, S. Montag, L. Rosen, S. Sunday, M. A. Vira, and L. R. Kavoussi, "The comparison of three renal tumor scoring systems: C-index, padua, and renal nephrometry scores," *Journal of endourology*, vol. 25, no. 12, pp. 1921–1924, 2011.

[178] A. Skalski, J. Jakubowski, and T. Drewniak, "Kidney tumor segmentation and detection on computed tomography data," in *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*. IEEE, 2016, pp. 238–242.

[179] G. Yang, J. Gu, Y. Chen, W. Liu, L. Tang, H. Shu, and C. Toumoulin, "Automatic kidney segmentation in ct images based on multi-atlas image registration," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 5538–5541.

[180] Q. Yu, Y. Shi, J. Sun, Y. Gao, Y. Dai, and J. Zhu, "Crossbar-net: A novel convolutional network for kidney tumor segmentation in ct images," *arXiv preprint arXiv:1804.10484*, 2018.

[181] M. Nolden, S. Zelzer, A. Seitel, D. Wald, M. Müller, A. M. Franz, D. Maleike, M. Fangerau, M. Baumhauer, L. Maier-Hein *et al.*, "The medical imaging interaction toolkit: challenges and advances," *International journal of computer assisted radiology and surgery*, vol. 8, no. 4, pp. 607–620, 2013.

[182] S. Singh and S. Krishnan, "Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks," *arXiv preprint arXiv:1911.09737*, 2019.

[183] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4148–4158.

[184] H. Handels, *Medizinische Bildverarbeitung: Bildanalyse, Mustererkennung und Visualisierung für die computergestützte ärztliche Diagnostik und Therapie*. Springer-Verlag, 2009.

[185] R. Trullo, C. Petitjean, B. Dubray, and S. Ruan, "Multiorgan segmentation using distance-aware adversarial networks," *Journal of Medical Imaging*, vol. 6, no. 1, p. 014001, 2019.

[186] A. E. Kavur, N. S. Gezer, M. Barış, P.-H. Conze, V. Groza, D. D. Pham, S. Chatterjee, P. Ernst, S. Özkan, B. Baydar *et al.*, "Chaos challenge–combined (ct-mr) healthy abdominal organ segmentation," *arXiv preprint arXiv:2001.06535*, 2020.

[187] Z. Wu, C. Shen, and A. v. d. Hengel, "Bridging category-level and instance-level semantic image segmentation," *arXiv preprint arXiv:1605.06885*, 2016.

[188] T. Ross, A. Reinke, P. M. Full, M. Wagner, H. Kenngott, M. Apitz, H. Hempe, D. M. Filimon, P. Scholz, T. N. Tran *et al.*, "Robust medical instrument segmentation challenge 2019," *arXiv preprint arXiv:2003.10299*, 2020.

[189] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 113–123.

[190] L. Heinrich, J. Funke, C. Pape, J. Nunez-Iglesias, and S. Saalfeld, "Synaptic cleft segmentation in non-isotropic volume electron microscopy of the complete drosophila brain," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 317–325.

[191] Y. Qin, K. Kamnitsas, S. Ancha, J. Nanavati, G. Cottrell, A. Criminisi, and A. Nori, "Autofocus layer for semantic segmentation," in *MICCAI*. Springer, 2018, pp. 603–611.

[192] Y. Xue, H. Tang, Z. Qiao, G. Gong, Y. Yin, Z. Qian, C. Huang, W. Fan, and X. Huang, "Shape-aware organ segmentation by predicting signed distance maps," *arXiv preprint arXiv:1912.03849*, 2019.

[193] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, and I. B. Ayed, "Boundary loss for highly unbalanced segmentation," *arXiv preprint arXiv:1812.07032*, 2018.

[194] S. Shit, J. C. Paetzold, A. Sekuboyina, A. Zhylka, I. Ezhov, A. Unger, J. P. Pluim, G. Tetteh, and B. H. Menze, "cldice–a topology-preserving loss function for tubular structure segmentation," *arXiv preprint arXiv:2003.07311*, 2020.

[195] J. H. Moltz, A. Hänsch, B. Lassen-Schmidt, B. Haas, A. Genghi, J. Schreier, T. Morgas, and J. Klein, "Learning a loss function for segmentation: A feasibility study," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 357–360.

[196] Z. Zhou, V. Sodha, M. M. R. Siddiquee, R. Feng, N. Tajbakhsh, M. B. Gotway, and J. Liang, "Models genesis: Generic autodidactic models for 3d medical image analysis," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 384–393.

[197] Q. Yu, D. Yang, H. Roth, Y. Bai, Y. Zhang, A. L. Yuille, and D. Xu, "C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4126–4135.

[198] F. Isensee, M. Schell, I. Pflueger, G. Brugnara, D. Bonekamp, U. Neuberger, A. Wick, H.-P. Schlemmer, S. Heiland, W. Wick *et al.*, "Automated brain extraction of multisequence mri using artificial neural networks," *Human brain mapping*, vol. 40, no. 17, pp. 4952–4964, 2019.

[199] G. Brugnara, F. Isensee, U. Neuberger, D. Bonekamp, J. Petersen, R. Diem, B. Wildemann, S. Heiland, W. Wick, M. Bendszus *et al.*, "Automated volumetric

assessment with artificial neural networks might enable a more accurate assessment of disease burden in patients with multiple sclerosis," *European Radiology*, pp. 1–9, 2020.

[200] F. Isensee and K. H. Maier-Hein, "Or-unet: an optimized robust residual u-net for instrument segmentation in endoscopic images," *arXiv preprint arXiv:2004.12668*, 2020.

List of Figures

List of Tables