# Pre-trained Models for nnUNet

by

Shivam Bajpai

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved February 2021 by the
Graduate Supervisory Committee:

Jianming Liang, Chair
Yalin Wang
Hemanth Kumar Demakethepalli Venkateswara

ARIZONA STATE UNIVERSITY

May 2021

# ABSTRACT

Image segmentation is one of the most critical tasks in medical imaging, which identifies target segments (e.g., organs, tissues, lesions, etc.) from images for ease of analyzing. Among nearly all of the online segmentation challenges, deep learning has shown great promise due to the invention of U-Net, a fully automated, end-to-end neural architecture designed for segmentation tasks. Recent months have also witnessed the wide success of a framework that was directly derived from U-Net architecture, called nnU-Net ("no-new-net"). However, training nnU-Net from scratch takes weeks to converge and suffers from unstable performance. To overcome the two limitations, instead of training from scratch, transfer learning was employed to nnU-Net by transferring generic image representation learned from massive images to specific target tasks. Although the transfer learning paradigm has proven a significant performance gain in many classification tasks, its effectiveness of segmentation tasks has yet to be sufficiently studied, especially in 3D medical image segmentation. In this thesis, first, nnU-Net was pre-trained on large-scale chest CT scans (LUNA 2016), following the self-supervised learning approach introduced in Models Genesis. Further, nnU-Net was fine-tuned on various target segmentation tasks through transfer learning. The experiments on liver/liver tumor, lung tumor segmentation tasks demonstrate a significantly improved and stabilized performance between fine-tuning and learning nnU-Net from scratch. This performance gain is attributed to the scalable, generic, robust image representation learned from the consistent and recurring anatomical structure embedded in medical images.

*To my grandmother, Deveshwari Bajpai and my parents, Ravindra and Rita Bajpai*

# ACKNOWLEDGEMENTS

First and foremost, I thank my family for their immense love and support throughout my academics. Thank you for motivating me in this journey. Without you, nothing would have been possible.

I would like to express my sincere gratitude to my advisor, Dr. Jianming Liang, for all his guidance and support throughout my thesis, and especially his confidence in me. Your insights pushed me to learn more and helped me in sharpening my skills. Further, I thank Dr. Yalin Wang and Dr. Hemanth Kumar Demakethepalli Venkateswara for accepting to be part of my thesis committee and reviewing my work towards betterment.

I would like to extend my gratitude to Fabian Isensee for his work and constant support. Further, I would like to thank the members of Dr. Liang's lab: Zongwei Zhou for his immense support in the experiments throughout my thesis; Vatsal Sodha for his guidance while working on Models Genesis; Fatemeh Haghighi and Mohammad Reza Hosseinzadeh Taher for their implementation of Semantic Genesis; Dr. Ruibin Feng, Md Mahfuzur Rahman Siddiquee, Diksha Goyal and Jiaxuan Pang for providing insightful suggestions and having discussions with me. I would also like to thank Soundararajan Velu, Liana Estes, and Ritesh Mishra for giving me confidence and motivation throughout my education.

This work has utilized the GPUs provided partially by the ASU Research Computing and partially by the Extreme Science and Engineering Discovery Environment (XSEDE) funded by the National Science Foundation (NSF) under grant number ACI-1548562.

This work is only the beginning of my journey.

# TABLE OF CONTENTS

Page

CHAPTER

APPENDIX

# LIST OF TABLES

## LIST OF FIGURES

Figure Page

# Chapter 1

# INTRODUCTION

## 1.1 Background

In past years, the increase in the number of digital images has attracted a lot of researchers to make scientific discoveries. Most of these discoveries leverage Image Processing to extract useful information from the image and utilize it in numerous tasks. One of these tasks is Image Segmentation, a process to divide an image into meaningful segments. In medical image analysis, these meaningful segments refer to the biologically relevant structures like tumors, organs, etc., making it easier for doctors in forming decisions. Therefore, developing superior methods for precise segmentation has become a hotspot of research in the medical community. Most of the online competitions thrive on finding these methods, either semi-automated or automated. In recent years, deep learning has become the mainstream approach to develop these methods. Within deep learning, Convolutional Neural Networks (CNNs) have achieved higher performance and have become a backbone for tasks like classification. With the help of downsampling in multiple steps, CNNs try to find the unique features in order to classify an image. To further improve the performance, various extensions of CNNs have been developed, like AlexNet, ResNet, DenseNet, etc.

However, in image segmentation, each pixel should be assigned a class label in order to detect the object of interest. This can be achieved with the help of upsampling operations. To do so, Ronneberger *et al.* (2015) have proposed U-Net architecture consisting of an encoder and decoder block. The encoder downsamples the image to find the unique features while the decoder upsamples these features and localizes the object of interest, resulting in a segmentation map.

In medical image analysis, U-Net, with 22,574 citations, has been the state-of-the-art method for segmentation tasks, even though numerous variations of U-Net have been developed to achieve better performance. Isensee *et al.* (2018) proposed the nnU-Net ("no-new-net") framework, based on U-Net architecture, surpassing most of the existing approaches on 23 public datasets in medical image segmentation challenges. The authors hypothesized that a "basic U-Net can outperform other architectures, given that the corresponding pipeline is designed adequately" (Isensee *et al.*, 2017a, p. 2). To design the corresponding pipeline, the nnU-Net framework adapts to a specific task based on dataset attributes. Further, a set of heuristic rules, combined with hardware constraints, determine the exact network topology, patch size, batch size, and image pre-processing. Despite the success of the nnU-Net framework, the framework has its own hassles causing unstable performance. After the thorough analysis of the framework, we observed the following disadvantages:

1. The framework utilizes learn from scratch strategy i.e. random initialization of weights while training the tasks with a small dataset. Further, training a specific task takes weeks to converge.

2. The framework formulates numerous specialized architectures due to the dependency on a specific dataset.

The above limitations suppress the performance of the nnU-Net framework on a specific task. To tackle the first limitations, we explored transfer learning. Further, we approach the second limitation by utilizing deeper architecture, like UNet++, to integrate into the nnU-Net framework.

Transfer Learning is used to initialize the starting point of a neural network to be trained on a specific task, with the parameters of a neural network already trained on a similar task. In the real world, we do not learn everything from scratch, and we instead utilize prior knowledge or what we know while learning new things. For example, if someone knows how to play the flute, he/she may be able to play the harmonica with little guidance. A similar concept has been used in the deep learning paradigm.

The traditional method to train a neural network requires a large amount of labeled data in order to achieve high performance. However, given the fact that data annotation is costly, acquiring such a dataset is tough. This trade-off between the amount of labeled data and performance can be bridged with the help of transfer learning. Pan and Yang (2010) defined transfer learning as:

> Given a source domain $\mathrm{D_S}$ and learning task $\mathrm{T_S}$, a target domain $\mathrm{D_T}$ and learning task $\mathrm{T_T}$ , transfer learning aims to help improve the learning of the target predictive function $\mathrm{f_T}(\cdot)$ in $\mathrm{D_T}$ using the knowledge in $\mathrm{D_S}$ and $\mathrm{T_S}$, where $\mathrm{D_S} \neq \mathrm{D_T}$ or $\mathrm{T_S} \neq \mathrm{T_T}$ (Pan and Yang, 2010, p. 3).

In the medical community, labeled data comes with high costs, while unlabelled data is generated constantly and exists in a relatively higher volume. In order to utilize the unlabelled data for the source task, we explored self-supervised learning. Self-supervised methods assist in the learning of generic visual features from images without utilizing human-annotated labels. This can be achieved with the help of pseudo labels extracted based on the attributes of an image. These pseudo labels help in learning the feature representation from unlabelled data, through training the network to learn the objective functions of the proxy tasks. Moreover, the pseudo labels provide supervision to the network from the data itself. The learned feature representation carries good semantic or structural meanings, further helping in the downstream task. Below are the advantages of transfer learning:

1. Even the reduced amount of data for the target task may provide good performance.

2. Results in a more effective and accurate model for the task at hand.

Even though transfer learning has been shown to improve the model for a specific task at hand, the disadvantage is the architecture depth for the target task. The architecture depth is unknown and has to be fixed for weight transfer.

## 1.2 Research Question

The above limitations of transfer learning and the nnU-Net framework gave rise to the following question: *Can we enhance the performance of the nnU-Net framework and make it more stable by advantageously integrating it with transfer learning?* The purpose of this work is based on answering this question. To do so, we started exploring different tasks utilized in the nnU-Net framework. The nnU-Net framework provides ten architectures for ten different tasks. With the source task and dataset in mind, we chose a specific architecture from the nnU-Net framework (discussed in the methodology section). Further, the nnU-Net framework determines numerous specialized architectures, and based on this observation, we will also address the following question: *Can we utilize one single architecture to learn multi-scale image features?*

## 1.3 Hypothesis

Pre-trained ImageNet models, proposed by Deng *et al.* (2009), have provided a significant boost in both natural and medical imaging applications. Further, Tajbakhsh *et al.* (2016) confirmed that the use of pre-trained models with sufficient fine-tuning is always equivalent to, or better than, training a model from scratch. While pre-trained ImageNet models are supervised and based on 2D natural images, Zhou *et al.* (2019) stated that for biomedical imaging applications, the models pre-trained on medical images can yield a more powerful target model than the models pre-trained on natural images, preserving 3D anatomical information. Due to the lack of labeled data in the medical community, Zhou *et al.* (2019) proposed a set of pre-trained models, named Models Genesis, which learn representations from large-scale medical images via self-supervision.

Based on the above exploration, we hypothesize that the performance of nnU-Net can be boosted significantly in two ways:

1. By employing transfer learning to nnU-Net: by transferring generic image representation learned from the massive images to a specific target task. To train the target task, we utilized the starting point from Models Genesis based on nnU-Net architecture.

2. By integrating an advanced segmentation architecture in the nnU-Net framework. We utilized UNet++ proposed by Zhou *et al.* (2018) in this work.

Our hypothesis is supported by the set of results showcased in the experimentation section.

## 1.4 Terminology

We define terminologies referred in this document as follows:

1. **Domain**: A feature space and a marginal probability distribution of the whole dataset define the domain. If two domains are similar enough, then they will have similar feature spaces and similar marginal distributions.

2. **Task**: A label space i.e. class labels and a learned function from training samples define a task. Different tasks may have different label spaces.

3. **Human-annotated Labels**: the ground truth labels defined by experts.

4. **Pseudo Labels**: defined from the data itself without any annotation cost.

5. **Proxy Task**: learns the predictive function using pseudo labels. The knowledge gained is used for the similar tasks at hand.

6. **Target Task**: computer vision applications utilizing the knowledge from the proxy task to evaluate the learned feature representations.

7. **Downstream Task**: computer vision applications utilizing human-annotated labels to evaluate the learned feature representation from pre-trained models. Downstream tasks may have less amount of training data.

# Chapter 2

# RELATED WORKS

## 2.1 Image Segmentation

Convolutional neural networks (CNNs) have become the core interest in deep learning for various problems, like image classification, speech recognition, etc. Due to the recent advancements in computing resources and data, CNNs are emerging swiftly within the computer vision community. Multiple variations of CNN's are introduced, like AlexNet, proposed by Krizhevsky *et al.* (2012); VGG-16, proposed by Simonyan and Zisserman (2015); GoogLeNet, proposed by Szegedy *et al.* (2014); and ResNet, proposed by He *et al.* (2015). The above variations demonstrate the importance of depth in a neural network. However, these architectures typically serve the purpose of classification on an image level, while the segmentation task requires classification at the pixel level, in order to localize an object within an image. To overcome this limitation, Ciresan *et al.* (2012) introduced a deep neural network to classify each pixel in an image with the help of the sliding window approach. Further, Long *et al.* (2015) proposed fully convolutional networks by replacing fully connected layers from deep neural networks with convolutional layers in order to produce a heatmap. Due to the low resolution of the output, in the fully convolutional networks, the segmentation map became fuzzy. Therefore, Ronneberger *et al.* (2015) extended fully convolutional networks by introducing upsampling layers in order to increase the resolution of the output and proposed an encoder-decoder architecture named U-Net. To get the segmentation map, the encoder extracts the features, then a decoder projects these features to a higher resolution.

U-Nets are widely used in medical image segmentation tasks because the introduction of skip connections recover full spatial resolutions in the output. However, U-Net architecture was introduced for medical image segmentation tasks in 2D while most of the medical images consist of 3D volumes. To tackle this problem, Milletari *et al.* (2016) introduced V-Net with residual blocks, in encoder and decoder, for 3D medical image segmentation tasks. At the same time, Çiçek *et al.* (2016) proposed 3D U-Net by replacing 2D operations in U-Net, with 3D operations, to segment the volumetric medical images. Further, multiple publications were introduced that required specialized architecture and training schemes to achieve desired performance on a specific dataset. Few of them are: (1) Isensee *et al.* (2017b) utilized and modified the U-Net architecture to process large 3D input blocks for brain tumor segmentation, (2) Isensee *et al.* (2017a) designed the architecture for cardiac disease assessment, (3) Li *et al.* (2018) proposed H-DenseUNet for liver and liver tumor segmentation, (4) Oktay *et al.* (2018) introduced Attention U-Net for pancreas segmentation. (5) Zhou *et al.* (2018) proposed UNet+ and UNet++ architecture by redesigning the skip pathways in the U-Net architecture to reduce the semantic gap between the encoder and decoder features. To address this issue within the above publications, Isensee *et al.* (2018) introduced the nnU-Net framework, which surpassed the existing approaches in the Medical Segmentation Decathlon (2019) challenge. Based on dataset properties, the framework decides the training scheme, such as input patch size, batch size, number of pooling layers, etc. On another note, as discussed in the introduction section, the nnU-Net framework has its own hassles. Further, we utilized the framework in this work to overcome the limitations.

## 2.2 Transfer Learning

We explored transfer learning to find the answer to the question discussed in the introduction section. Deng *et al.* (2009) proposed pre-trained ImageNet models, giving a significant boost in the performance of both natural and medical imaging applications. Further, Tajbakhsh *et al.* (2016) confirmed that the use of pre-trained models with sufficient fine-tuning will always be equivalent to or better than training a model from scratch. However, the ImageNet dataset is large scale and contains 1.3 million labeled natural images annotated by humans. Additionally, ImageNet models have been solved in 2D while the medical modalities like CT and MRI belong to 3D imaging tasks. Due to this reason, 3D imaging tasks have to be solved in 2D to utilize pre-trained ImageNet models, hence losing 3D anatomical information. Therefore, Carreira and Zisserman (2017) proposed I3D models, expanded in 3D, pre-trained on Kinetics dataset based on the temporal video. Due to the domain gap in temporal video and 3D medical images, Gibson *et al.* (2018) introduced model zoo in NiftyNet for specific applications; Chen *et al.* (2019b) proposed Med3D pre-trained models by jointly training eight annotated medical datasets. The above publications require a large labeled dataset for pre-trained models. Acquiring such a large dataset in medical images is expensive and time-consuming. To avoid this limitation, various self-supervised methods have been introduced. Zhang *et al.* (2016) introduced a colorization proxy task where a grayscale image is mapped to color value output using CNN architecture. In addition, Pathak *et al.* (2016) suggested context encoders where encoder-decoder architecture reconstructs the missing region in an input image. Further, Noroozi and Favaro (2017) put forward a method to learn visual representation by solving jigsaw puzzles in an unsupervised fashion. Similarly, Gidaris *et al.* (2018) advanced unsupervised representation learning by predicting image rotations.

However, the above publications were based on the context of natural images, though natural images are statistically different from medical images. Further for medical applications, Ross *et al.* (2018) proposed colorization as a proxy task for colonoscopy images; Chen *et al.* (2019a) designed image restoration as a proxy task, where small windows within the image were shuffled for the model to learn the original image; Zhuang *et al.* (2019) introduced the proxy task by recovering the refactored rubik's cube. These methods were developed individually for specific target tasks with limited generalizability over multiple tasks.

To tackle this limitation, Zhou *et al.* (2019) proposed a self-supervised learning method by utilizing the properties of medical images, namely, a collection of pre-trained models called Generic Autodidactic Models, nicknamed Models Genesis. The author, Zongwei Zhou, was presented the Young Scientist Award in 2019 by MICCAI, the top conference in medical imaging, and MedIA best paper award. Zhou *et al.* (2019) stated that "models pre-trained on medical images can yield a more powerful target model than the models pre-trained on natural images". Models Genesis is the first pre-trained model for open science in medical images.

# Chapter 3

# METHODOLOGY

To demonstrate the boost in the performance, we will be utilizing the pre-trained models from Models Genesis in the nnU-Net framework. First, we will delve into a briefing of the nnU-Net framework and Models Genesis.

## 3.1 The nnU-Net Framework: Adaptive Framework for Medical Image Segmentation

The nnU-Net framework is an open-source algorithm that can be used out-of-the-box for multiple segmentation tasks. The framework has been designed based on the dataset properties for different segmentation tasks. The traditional approach (Figure 3.1) requires the knowledge of architecture and training parameters beforehand. In contrast, the nnU-Net framework utilizes generic (standard) U-Net architecture and determines training parameters based on the available knowledge of a specific dataset in order to design the algorithm pipeline.

The framework is developed based on datasets provided by the Medical Segmentation Decathlon (2019) challenge. To determine the optimized pipeline, the framework has been divided into multiple components as shown in Figure 3.2.

A brief description of individual components are depicted below:

1. **Data Fingerprints**: This component of the framework accumulates the dataset properties like image size, image spacing, modality, class label, etc. Further, the framework utilizes these attributes in the pre-processing steps.

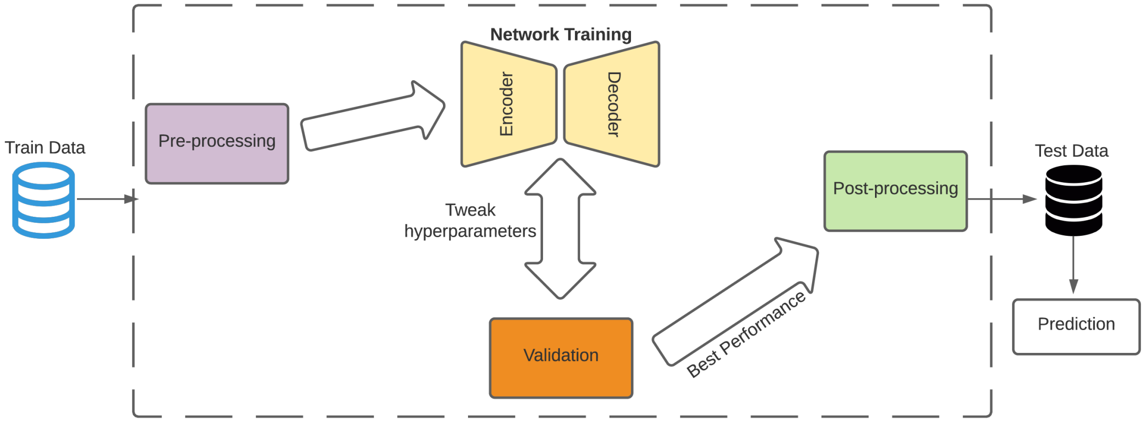

Figure 3.1: Traditional Approach where known hyper-parameters and fixed architecture configuration is used, Irrelevant to dataset attributes.

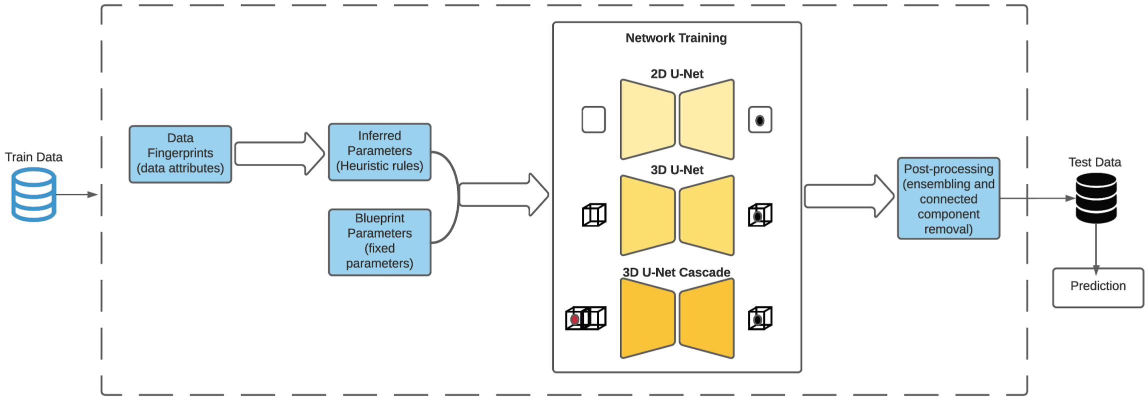
Train Data
Data Fingerprints (data attributes)
Inferred Parameters (Heuristic rules)
Blueprint Parameters (fixed parameters)
Network Training
2D U-Net
3D U-Net
3D U-Net Cascade
Post-processing (ensembling and connected component removal)
Test Data
Prediction

Figure 3.2: Automated design of the nnU-Net Framework: Inferred parameters heuristic rules operate on data fingerprint. Blueprint parameters and Inferred parameter together make the input for network training. Three network architectures are trained based on the input from the framework in a 5-fold cross-validation way. Empirical parameters choose the optimal architecture based on the performance of validation data.

2. **Blueprint Parameters**: The parameters decide the architecture template, training schedule, and inference choices. The framework employs original U-Net architecture proposed by Ronneberger *et al.* (2015) and Çiçek *et al.* (2016). Large patch size is favored over batch size. The networks are trained for 1000 epochs with 250 iterations within each epoch. The Sum of cross-entropy and dice loss is used as a loss function. In order to handle class imbalance, each

batch includes one patch from the foreground class, and another one is randomly sampled. Various data augmentation strategies are applied during training. Within the inference step, the sliding window approach is used for prediction.

3. **Inferred Parameters**: An individual image, with its mean and standard deviation, is normalized using z-score normalization, excluding CT images. For CT images, 0.5 and 99.5 percentiles of the foreground pixels are clipped, followed by the global normalization scheme, using standard deviation and the mean of the whole dataset. To deal with heterogeneous voxel spacing, images are resampled to the target spacing using either third-order spline, linear or nearest-neighbor interpolation. The patch size is initialized as a median shape after resampling. Based on the patch size, the architecture is configured by determining the number of downsampling layers (until the feature map reduced to 4 voxels). The patch size and architecture topology is adjusted in an iterative process until the GPU memory budget is met.

4. **Empirical Parameters**: The component ensembles and selects the best model configuration. The framework uses a fivefold cross-validation strategy to train individual configurations (2D U-Net, 3D U-Net, and 3D U-Net Cascade) and selects either single or ensemble of two U-Net configurations. In post-processing, the framework decides whether or not to remove all, but the largest connected component.

Based on the results reported by Isensee *et al.* (2018), 3D U-Net is the favored option for most of the tasks. In this work, we experimented and demonstrated performance gain using 3D U-Net configuration from the nnU-Net framework alone.

## 3.2 Models Genesis: A Self-supervised Framework for 3D Medical Image Analysis

Fine-tuning pre-trained ImageNet models have become the de facto standard in classification and segmentation tasks. Even though they provide a boost in the performance, they have the following limitations:

1. Use 2D images during training while, in medical imaging applications, most of the images are 3D, and solving 3D imaging target tasks in 2D might lose contextual information, resulting in low performance.

2. Are supervised and use a large amount of labeled data. In medical imaging applications, acquiring such an amount of labeled data is expensive.

3. Are trained on natural images.

To tackle the above limitations, Zhou *et al.* (2019) proposed a set of pre-trained models named Models Genesis. Models Genesis learns the generic anatomical patterns by utilizing a series of self-supervised strategies. As shown in Figure 3.3, the predictive function is trained to learn the reconstruction of the original image from the transformed image using an encoder-decoder architecture.

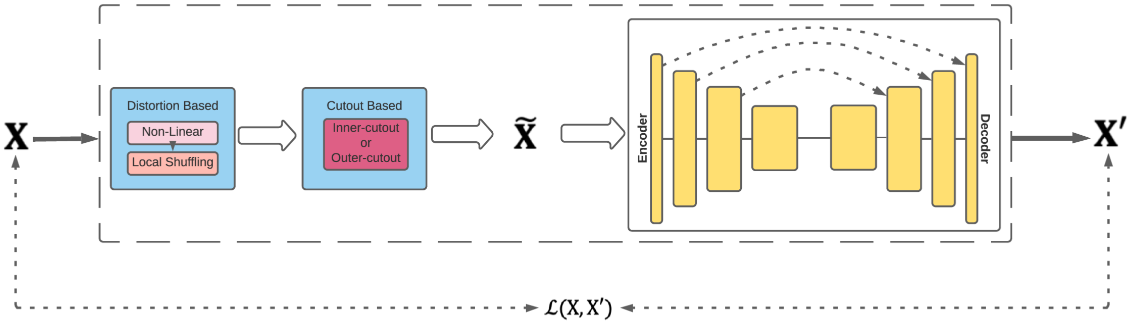

Figure 3.3: The original image is transformed using distortion and cutout-based methods. The network learns the generic anatomical representation from the transformed image by recovering the original image. The network is trained to minimize the L2 distance between the prediction and ground truth.

The performance evaluation depicted in the paper shows the significance of Models Genesis and its transferability beyond organs, diseases, and modalities in medical imaging applications. In contrast with the ImageNet pre-trained models, Models Genesis has the following advantages:

1. Are self-supervised and utilizes most of the unlabelled data without any human annotation.

2. Solves 3D imaging tasks in 3D instead of 2D, preserving rich 3D anatomical patterns.

3. Depicted in Zhou *et al.* (2019), the exceptional results show that the pre-trained models on medical images are more favorable and positively influence the target task as opposed to the pre-trained models on natural images.

In the training process of the proxy task, 3D patches from the image are extracted to go through transformation strategies. The 3D patches extracted from the images are used as the ground truth, while the transformed patches are used as the input to the architecture for the reconstruction task. To learn the generic representations, Models Genesis utilizes the below self-supervised transformation strategies:

1. **Non-Linear Transformation**: Utilizes Bezier curve as transformation function on the input patch. It enables the model to learn organ appearances and intensity mapping.

2. **Local pixel shuffling**: Samples a random window from the patch and then shuffles the pixels contained in it. This strategy helps the model to learn the texture and edges.

3. **Cutout methods**: Implements outer-cutout and inner-cutout methods to learn the context present in the patch. For outer-cutout, random windows of different

sizes are superimposed together, followed by randomly assigning the pixel values outside the window. Therefore, the outer-cutout learns the global geometry and spatial layout of the organs in the patch. For inner-cutout, pixel values inside the window are assigned with the constant value in order to learn the local context of organs in the patch.

## 3.3 Integrating Models Genesis in the nnU-Net Framework

The nnU-Net framework is designed for 3D imaging tasks, while Models Genesis, being a self-supervised learning method, has out-performed 3D models trained from scratch. With the widespread success of the nnU-Net framework, we reproduced the results on each task and found that the performance of the framework is unstable due to the random initialization of weights. Based on the above observation, we demonstrate that initializing the starting point of the nnU-Net architecture from Models Genesis will boost the nnU-Net performance: especially for those applications with limited annotation. This gain is attributed to learning representation from large-scale medical images via self-supervision.

The nnU-Net framework trains three variations of architecture for each task i.e. 2D U-Net, 3D U-Net, and 3D U-Net Cascade as shown in Figure 3.2. Based on the results reported by the author, 3D U-Net is the most favorable option for 3D imaging tasks. Due to this reason, we utilized the 3D U-Net architecture, extracted from the nnU-Net framework, and demonstrated that the performance can be enhanced by fine-tuning Models Genesis for lung tumor, liver organ, and liver tumor segmentation tasks. So far, our proxy task utilizes only the LUNA16 dataset, implying that Models Genesis never sees any of the images from the target tasks. To learn the generic representation in the proxy task, we first extracted the fixed-sized patches from the LUNA16 dataset followed by z-score normalization based on the mean and standard deviation of the whole dataset. After pre-processing, we transformed the patches using the transformation strategies of Models Genesis. The transformed patch is then fed to the nnU-Net architecture in order to reconstruct the original patch. The network is trained to minimize the L2 distance between the predicted patch X′ and ground truth X (original patch without transformation).

$$L(X) = ||f(X') - X||_2^2 \tag{3.1}$$

When the network converged, we fine-tuned Models Genesis on the target task. The training and the fine-tuning process are depicted in Figure 3.4. The main question that arose was: *What architecture configuration can we utilize to train Models Genesis?*. We further examine the details in the next section.

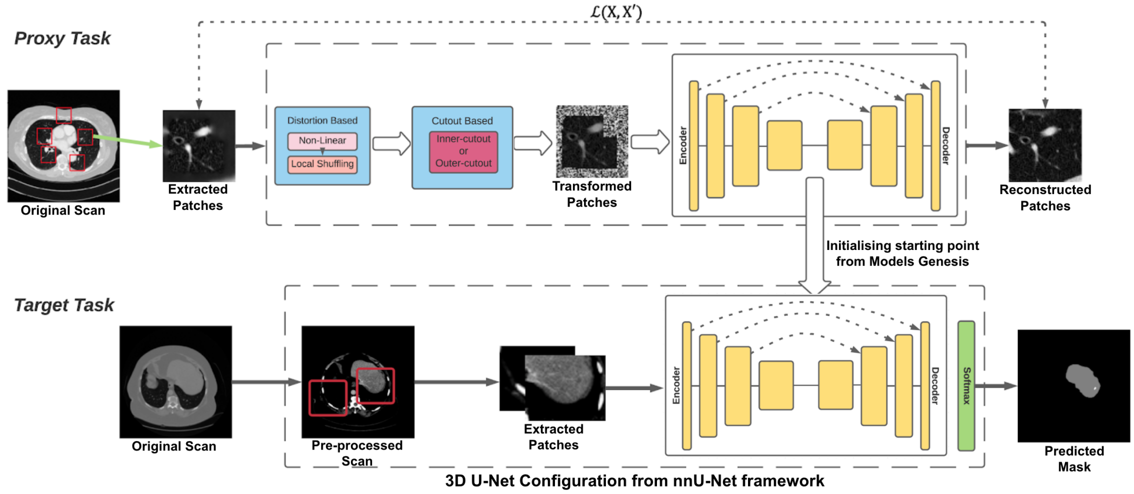

Figure 3.4: Using 3D U-Net architecture configuration from the nnU-Net Framework, Models Genesis (top: Proxy Task) is trained. In the target task, we initialize the starting point of the nnU-Net framework from Models Genesis.

The proxy task utilizes the LUNA16 dataset which was released with the motive to develop computer algorithms for lung cancer screening. To find the optimal architecture to train our proxy task, we first explored the differences between the ten tasks and their specific architectures (Figure 4.1). We observed that the Liver (Organ and Tumor) and Lung Tumor segmentation tasks use the same configuration with five layers in the encoder, one layer in a bottleneck, and five layers in the decoder. Apart from that, our proxy task dataset (LUNA16) provided by Setio *et al.* (2017) was released with a similar motive as the lung tumor dataset provided by Medical Segmentation Decathlon (2019) challenge. Due to this reason, we used the architecture configuration of the lung tumor segmentation task which was the same as the liver tumor segmentation task to train Models Genesis.

## 3.4 Advancing Segmentation Architecture in the nnU-Net Framework

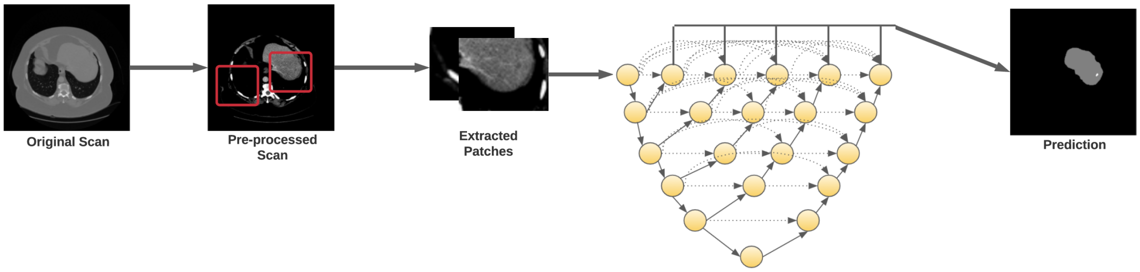

Figure 3.5: 3D U-Net replaced by UNet++. The convolution blocks use the same configuration of 3D U-Net architecture obtained from the liver and liver tumor segmentation task pipeline in the nnU-Net framework.

The nnU-Net framework determines numerous specialized architectures based on U-Net. Even though the framework shows promising results, no study has been done to test the impact of other deep architectures. We demonstrate that, by integrating deeper architecture in the framework, the performance on the specific task is enhanced. We demonstrated our hypothesis depicted in Figure 3.5, through refactoring the generic U-Net (Ronneberger *et al.* (2015)) architecture to UNet++ (Zhou *et al.* (2018)) for the liver organ and tumor segmentation tasks. Skip pathways in U-Net architecture connect the feature maps between encoder and decoder, directly resulting in fusing semantically dissimilar features. To avoid this, Zhou *et al.* (2018) proposed a deeply-supervised encoder-decoder network, where the encoder and decoder sub-networks are connected through a series of dense, nested skip pathways.

The redesigned skip pathways reduce the semantic gap between the features of the encoder and decoder sub-networks. We implemented UNet++, to learn multi-scale image features, in the nnU-Net framework based on the liver organ and tumor segmentation architecture configuration i.e. backbone with five layers in the encoder and one layer in the bottleneck. Our experiment shows that integrating deeper architecture can further boost the performance of a specific task. Utilizing a generic UNet++ architecture eliminates the need to ensemble numerous specialized architecture for a specific task.

# Chapter 4

# EXPERIMENTS AND RESULTS

## 4.1 Implementation

### *4.1.1 Dataset*

In this work, we have utilized the datasets provided by Medical Segmentation Decathlon (2019) in the target task. The challenge spans over ten datasets (Table 4.1) belonging to different modalities. For our proxy task, we have utilized the LUNA16 dataset provided by Setio *et al.* (2017). This dataset spans over 888 Chest CT scans.

### *4.1.2 Architecture Differences Between Tasks*

The framework configures ten 3D U-Net architectures for ten datasets (Fig. 4.1). All the architectures utilize a generic template with two convolution blocks within each layer. The difference in the architectures lies in the number of layers and the kernels within each layer. Figure 4.1 shows that most of the architectures have five layers in the encoder, one layer in a bottleneck, and five layers in the decoder. Hippocampus dataset folows shallow architecture as depicted in Figure 4.1(a). Lung, Liver, Brain, Heart, Pancreas, Hepatic vessel, Spleen, and Colon dataset follow same depth in the architecture as determined by the nnU-Net framework (Figure 4.1(b)). Even though they are similar, the prostate, pancreas, spleen, and colon differ in the first layer itself. The architecture determined for the Prostate dataset is the deepest, depicted in Figure 4.1(c). The architecture of the Brain Tumor Segmentation task

| **Dataset** | **Modalities** | **Total Classes** | **Number of Samples** |
|---|---|---|---|
| Brain tumor | MRI (T1, T1c,T2, FLAIR) | 3 | 484 |
| Heart | MRI | 1 | 20 |
| Liver | CT | 2 | 131 |
| Hippocampus | MRI | 2 | 260 |
| Prostate | MRI(T2, ADC) | 2 | 32 |
| Lung | CT | 1 | 63 |
| Pancreas | CT | 2 | 282 |
| Hepatic Vessel | CT | 2 | 303 |
| Spleen | CT | 1 | 41 |
| Colon | CT | 1 | 126 |

Table 4.1: Properties of datasets provided by Medical Segmentation Decathlon (2019).

looks similar to lung and liver architecture, however, the difference lies in the modality of these two tasks. Additionally, the Brain Tumor segmentation task belongs to the MRI domain and utilizes four modalities (T1, T1c, T2, and FLAIR) as input to the U-Net architecture, while lung and liver tasks belong to the CT domain with one modality in the input patch. Due to this reason, the transfer of weights in the first layer from Models Genesis becomes impossible. Further, we will show the impact of Models Genesis on all the target tasks.

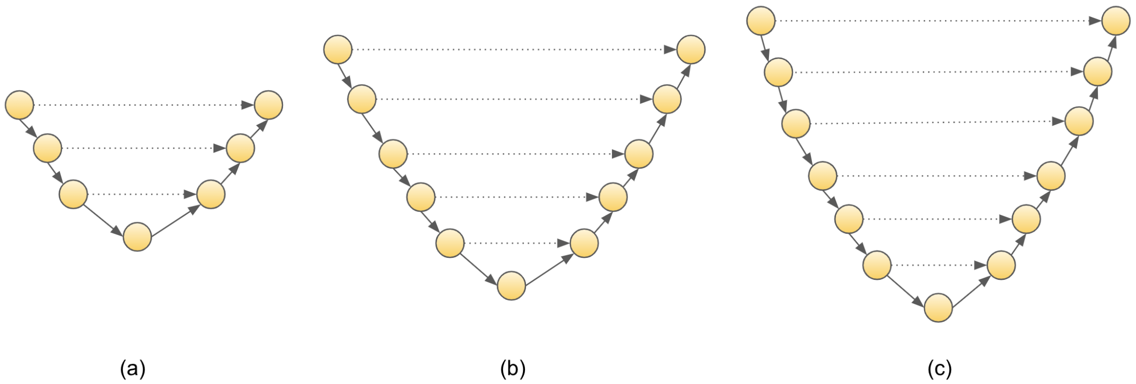

Figure 4.1: The depth of the architectures determined by the nnU-Net framework. (a) Hippocampus architecture. (b) Brain. Heart, Liver, Lung, Pancreas, Hepatic vessel, Spleen, and Colon architecture. (c) Prostate architecture. Note: All the architectures are determined by the nnU-Net framework.

### *4.1.3 Proxy Task*

We trained Models Genesis using the patches extracted from 445 CT scans of the LUNA16 dataset and used 178 CT scans for validation. The remaining 265 scans were used for testing purposes in order to evaluate Models Genesis. We used the patch size of 64x64x32 as input to the 3D U-Net architecture. Note that, no human annotation is utilized in our proxy task. The architecture configuration is derived based on liver and lung datasets in the nnU-Net framework.

### *4.1.4 Target Tasks*

The datasets in Table 4.1 correspond to the following segmentation tasks:

1. **Brain Tumor Segmentation**: The target task contains 484 training and 266 testing cases with the objective to segment 3 classes i.e. edema, active tumor, and necrosis.

2. **Heart Segmentation**: This target task contains 20 training and 10 testing cases with the objective to segment the left ventricle.

3. **Liver and Liver Tumor Segmentation**: The target task contains 131 training and 70 testing cases with the objective to segment liver and liver tumors.

4. **Hippocampus Segmentation**: The target task contains 263 training and 131 testing cases with the objective to segment two neighbor small structures i.e. anterior and posterior hippocampus.

5. **Prostate Segmentation**: The target task contains 32 training and 16 testing cases with the objective to segment prostate central and peripheral zone.

6. **Lung Tumor Segmentation**: The target task contains 64 training and 32 testing cases with the objective to segment lung tumors.

7. **Pancreas and Pancreas Cancer Segmentation**: The target task contains 282 training and 139 testing cases with the objective to segment pancreas organ and pancreas cancer.

8. **Hepatic Vessel and Tumor Segmentation**: The target task contains 303 training and 140 testing cases with the objective to segment hepatic vessels and tumors.

9. **Spleen Segmentation**: The target task contains 41 training and 20 testing cases with the objective to segment the spleen organ.

10. **Colon Cancer Segmentation**: The target task contains 126 training and 64 testing cases with the objective to segment colon cancer.

### 4.2 Hyper-parameters

In our proxy task, all the patches were normalized using z-score normalization with a mean of -775.8 and a standard deviation of 251.9. These patches were then used as the input to 3D U-Net architecture. The mean squared error (L2 norm) was used as the loss function. The stochastic gradient descent method was used as an optimizer with a learning rate of 1e-1. The learning rate reduces by the use of a learning rate scheduler if validation loss does not decrease after certain epochs. In the target task, we used the same settings for the hyper-parameters as pre-defined by the nnU-Net framework.

## 4.3 Models Genesis Results

Once the Models Genesis is trained, we evaluate it on the patches extracted from the scans reserved for testing purposes. None of the patches from the test set are seen by the model. Figure 4.2 shows the reconstructed patches by Models Genesis and confirms that Models Genesis, indeed, learns the anatomical patterns from the large-scale images via self-supervision.

## 4.4 Target Task Results

To evaluate our Models Genesis, we fine-tuned it on liver organ, liver tumor, and lung tumor segmentation tasks via transfer learning. We fine-tuned all the layers from encoder and decoder blocks in the above segmentation tasks. The weights were transferred for all except the last layer from the pre-trained model. We compared the fine-tuned Models Genesis results on the target tasks with training from scratch. Table 4.2 and 4.3 shows the stable and enhanced results on liver and liver tumor segmentation tasks. Further, Models Genesis achieved the first rank in the liver tumor segmentation task on the challenge leaderboard.

Table 4.4 and 4.5 shows lung tumor segmentation performance using Models Genesis and using random initialization. The enhanced performance highlights the significance of Models Genesis and the prior knowledge injected into the target task.

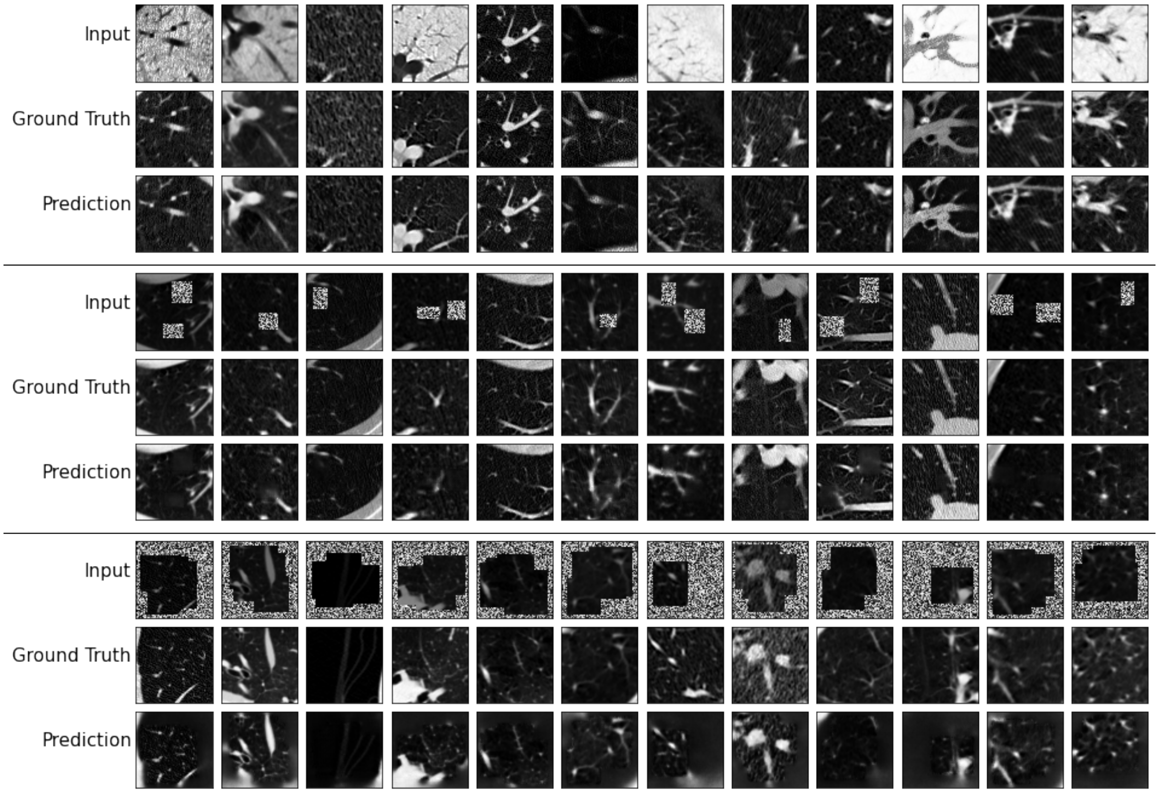

Figure 4.2: Qualitative results of Models Genesis on the test set. Once Models Genesis is trained, we evaluate it on the test set using the patches reserved for testing purposes.

| **Proxy Task** | **Purpose** | **Liver** | **Tumor** |
|---|---|---|---|
| Scratch | Validation | 96.19 | 63.37 |
| Models Genesis | Validation | 96.19 | **65.52** |

Table 4.2: Liver and Liver Tumor Segmentation: Validation. Fine-tuning Models Genesis outperforms the nnU-Net framework trained from scratch. Scores depicted are Dice scores. Note: The best result is denoted in bold.

| **Proxy Task** | **Purpose** | **Liver** | **Tumor** |
|---|---|---|---|
| Scratch | Test | 95.75 (Rank: 1) | 75.97 (Rank: 5) |
| Models Genesis | Test | 95.72 (Rank: 2) | **77.50 (Rank: 1)** |

Table 4.3: Liver and Liver Tumor Segmentation: Test Set. Fine-tuning Models Genesis outperforms the nnU-Net framework trained from scratch. Scores depicted are Dice scores. Note: The best result is denoted in bold. Rank resembles the challenge leaderboard ranking.

| **Proxy Task** | **Purpose** | **Tumor** |
|---|---|---|
| Scratch | Validation | 69.5 ± 1.13 |
| Models Genesis | Validation | **71.8 ± 1.4** |

Table 4.4: Lung Tumor Segmentation: Validation Set. Fine-tuning Models Genesis outperforms the nnU-Net framework trained from scratch. Scores depicted are Dice scores. Note: The best result is denoted in bold.

| **Proxy Task** | **Purpose** | **Tumor** |
|---|---|---|
| Scratch | Test | 73.97 (Rank: 5) |
| Models Genesis | Test | **74.54 (Rank: 3)** |

Table 4.5: Lung Tumor Segmentation: Test Set. Fine-tuning Models Genesis outperforms the nnU-Net framework trained from scratch. Scores depicted are Dice scores. Note: The best result is denoted in bold. Rank resembles the challenge leaderboard ranking.

## 4.5 Qualitative Results of Target Task

As shown in Figure 4.3 and Figure 4.4, Models Genesis can effectively segment and find the liver tumor and lung tumor region in the CT scans while the nnU-Net framework standalone misses those regions or does not segment it efficiently.

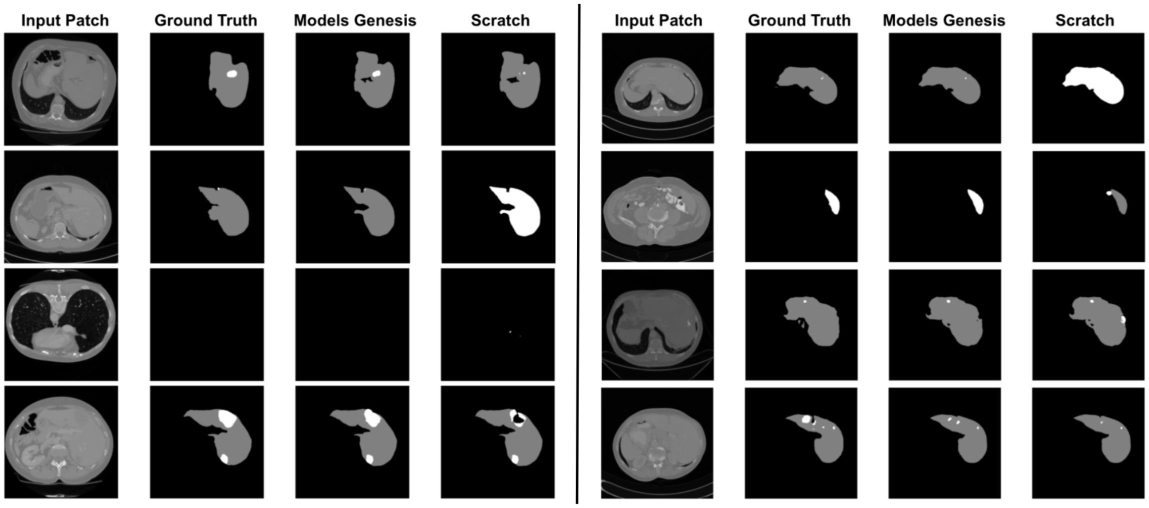

Figure 4.3: Qualitative results of Liver and Liver Tumor Segmentation Task. Once the Models Genesis is trained, we fine-tune it on the target task.

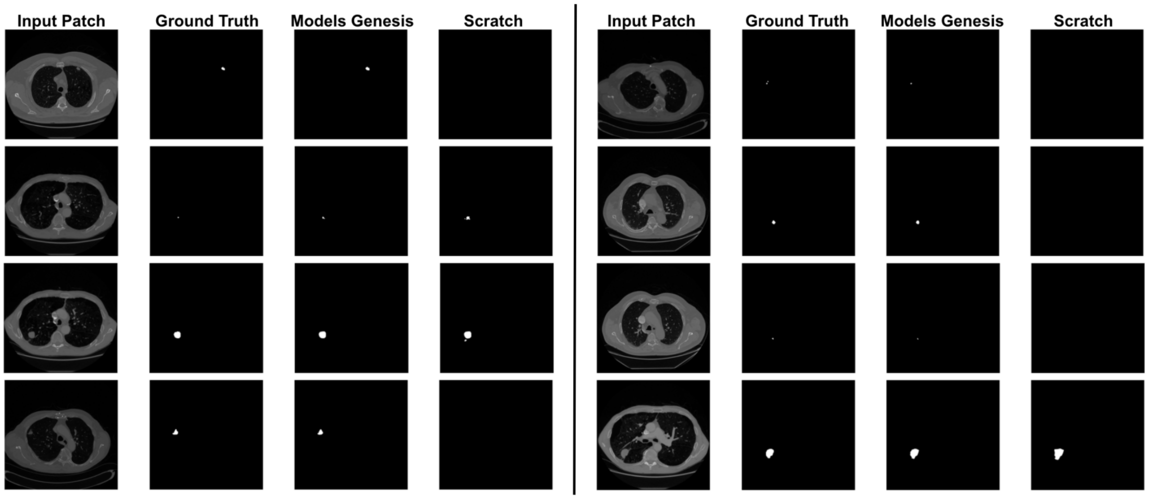

Figure 4.4: Qualitative results of Lung Tumor Segmentation Task. Once the Models Genesis is trained, we fine-tune it on the target task.

## 4.6 Results of UNet++ in the nnU-Net Framework

We replaced U-Net architecture in the nnU-Net framework with UNet++. For implementation, we use the same kernel shapes and layers from liver architecture as determined by the nnU-Net framework. Additionally, encoder-decoder sub-networks were injected as depicted in Zhou *et al.* (2018). The architecture contains five layers in the encoder, one layer in a bottleneck, and five layers in the decoder. The implementation is depicted in Figure 3.5. Table 4.6 shows the gained performance using UNet++ architecture instead of U-Net in the nnU-Net framework. The U-Net++ branch refers to encode-decoder sub-network predictions (Figure 4.5). The performance of UNet++ on the liver organ is similar to U-Net architecture while there is a significant improvement in liver tumor scores. As the encoder-decoder sub-network deepens the performance improves. In this way, UNet++ learns multi-scale image features. This experiment demonstrates that using advanced segmentation architecture in the nnU-Net framework can further improve the performance on s specific task.

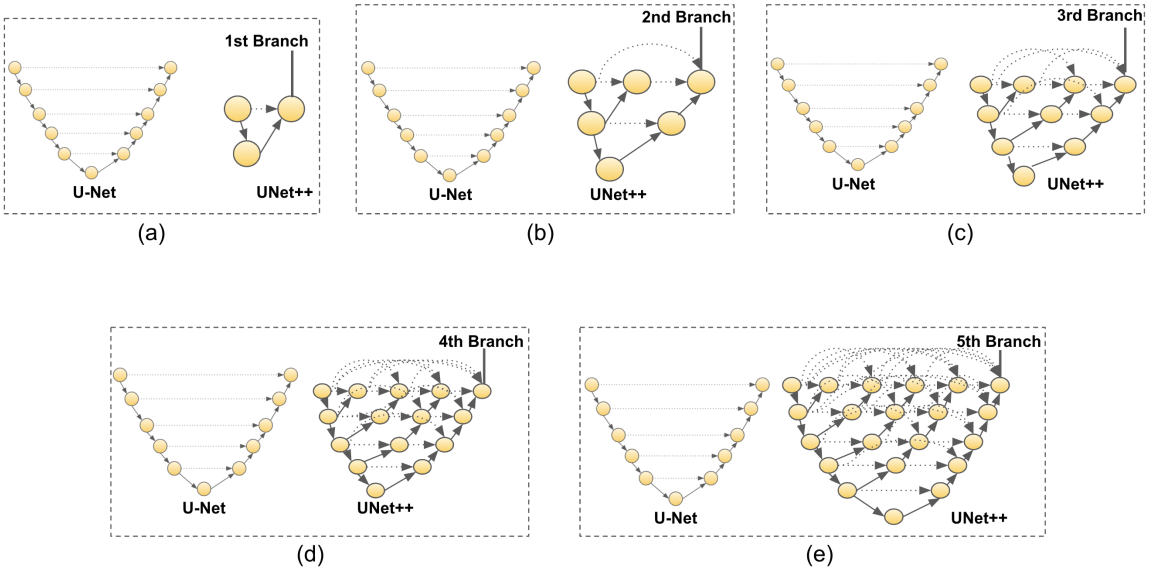

Figure 4.5: Multiple Branches of UNet++ Architecture. We compare individual branches in our experiment with U-Net architecture in the nnU-Net framework.

| **Experiment** | **Branch** | **Figure** | **Liver** | **Tumor** |
|---|---|---|---|---|
| U-Net | - | - | **96.18** | 63.37 |
| UNet++ | 1st branch | 4.5(a) | 89.96 | 38.03 |
| UNet++ | 2nd branch | 4.5(b) | 94.68 | 59.66 |
| UNet++ | 3rd branch | 4.5(c) | 95.78 | 65.83 |
| UNet++ | 4th branch | 4.5(d) | 96.03 | 66.02 |
| UNet++ | 5th branch | 4.5(e) | 96.11 | **66.25** |

Table 4.6: Liver and Liver Tumor Segmentation. The pipeline determined by the nnU-Net framework is used in the experiments. The only difference is in segmentation architecture. Training is done using random initialization of weights. Scores depicted are Dice scores. Note: The best result is denoted in bold.

## 4.7 Qualitative Results of UNet++ Integrated Into the nnU-Net Framework for the Liver Tumor Segmentation Task

As shown in Figure 4.6, UNet++ architecture in the nnU-Net framework can effectively segment and find the liver tumor region in the CT scans while U-Net architecture in the nnU-Net framework misses those regions or does not segment it efficiently.

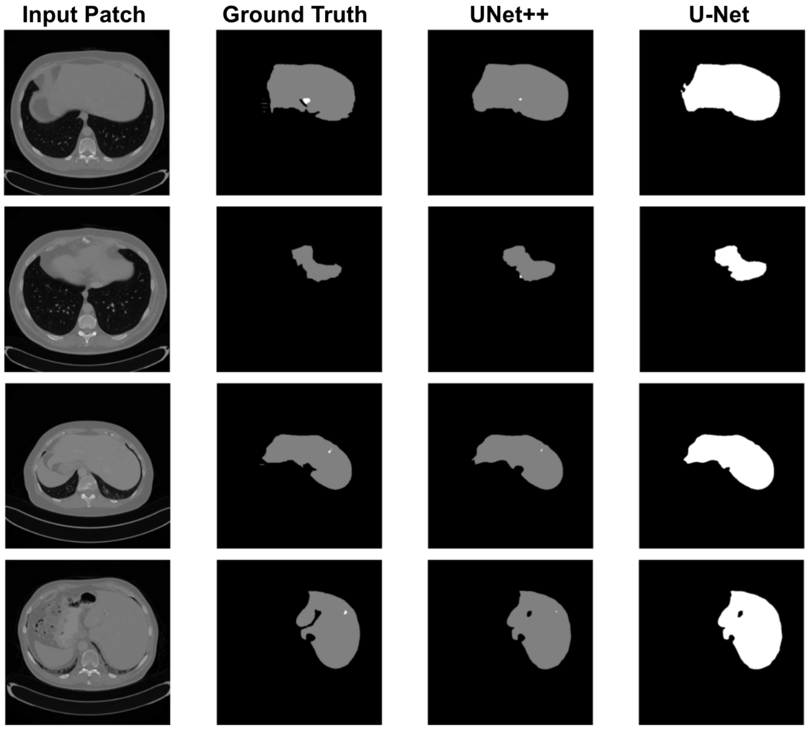

Figure 4.6: Qualitative results of Liver Tumor Segmentation Task using UNet++ in the nnU-Net framework. The results for UNet++ are based on the best branch i.e. 5th branch as depicted in Figure 4.5(e).

# Chapter 5

# DISCUSSION

## 5.1 Target Task Results on the Other Datasets in CT Domain

Even though there was an architecture difference between Models Genesis and the rest of the tasks belonging to the CT domain, we still evaluated the impact of the weight transfer on them. Pancreas, Spleen, and Colon architecture's first layer is different from the architecture of our proxy task. Figure 5.1 shows that the target task performance using the starting point from Models Genesis is similar to training from scratch. The results show the importance of the initial layers of the pre-trained model. Due to no weight transfer for them, the architecture trains similar to training from scratch.

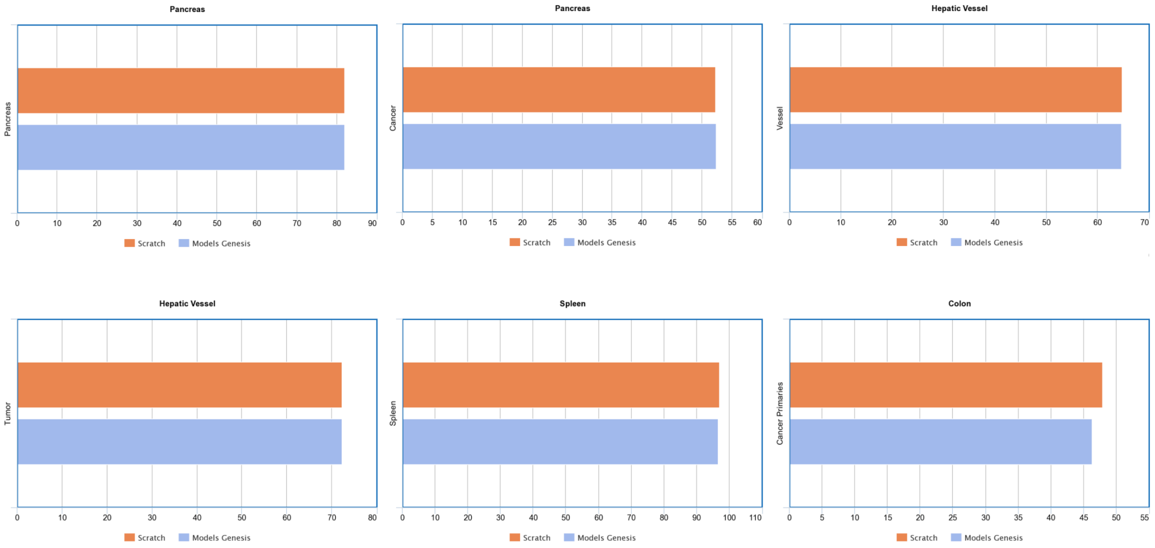

Figure 5.1: X-axis denotes Dice scores on the target task. We evaluated Models Genesis trained using Lung architecture and the LUNA16 dataset on the target tasks. The architecture of the target task is determined by the nnU-Net framework.

## 5.2 Do We Still Need Multiple Architectures?

The multiple architectures determined by the nnU-Net framework make it impossible to initialize the starting point from Models Genesis on the target task. Due to the variations in the architecture, the prior knowledge gained by Models Genesis comes to no use in the target task performance. Due to this, the following question arises: *Can we use a generic architecture for all the tasks?*

As shown in the experiment section, the architecture of the target tasks in the CT domain shares the same kernels in all, except the first layer. The above observation implies that the proxy task architecture is close enough to the rest of the target task architectures in the CT domain, however, the first layer difference contributes more to the training from scratch. Hence, replacing the first layer in the target tasks architecture with Models Genesis architecture will be required to utilize the prior knowledge gained by Models Genesis. Further, integrating UNet++ in the nnU-Net framework helps in learning the multi-scale image features and eliminates the need for multiple architectures for a specific task. This implementation can be extended via implementing UNet++ to learn multiple features to span across multiple tasks. In the future, we will be working on this hypothesis in order to create a generic architecture for multiple tasks, further utilizing the prior knowledge gained by the proxy task.

## 5.3 Cross Domain Transfer

Transfer learning positively influences the performance of the target task in the similar domain with the similar architecture of the proxy task. We have trained our proxy task using the LUNA16 dataset belonging to the CT domain. The trained Models Genesis does not seem to provide significant improvement for the target tasks in the MR domain.

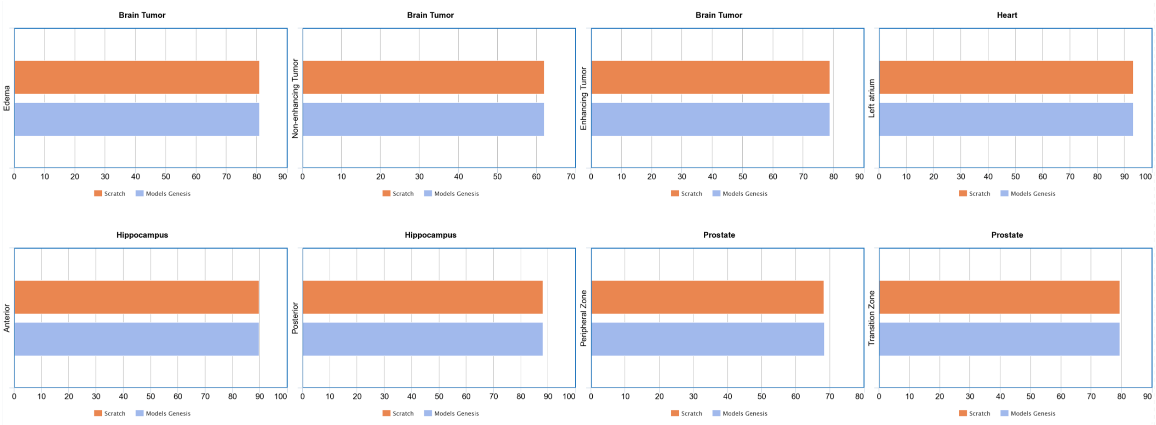

Figure 5.2: X-axis denotes Dice scores on the target task. We evaluated Models Genesis trained using Lung architecture and LUNA16 dataset on the target tasks, belonging to the MR domain. The architecture of the target task is determined by the nnU-Net framework.

Figure 5.2 shows that Models Genesis gives similar performance as training from scratch. The Brain tumor dataset has four modalities in each MR scan which causes the input patch for the architecture to have four sequences. On the other hand, Models Genesis is trained using only one sequence based on the LUNA16 dataset. Due to this reason, the weight transfer for the first layer does not initiate. The initial layers of the pre-trained model make a considerable impact on the target task. Hence, the architecture trains similar to the randomly initialized model. We found similar observations for the Hippocampus segmentation task. The architecture for the same is shallow with only three layers in the encoder and one layer in the bottleneck while the Genesis architecture contains five layers in the encoder and one layer in the bottleneck. This creates a huge difference in the architectures of proxy and target tasks, leading to performance similar to training from scratch.

The other reason for no improvement is the domain difference between the proxy task dataset (LUNA16) and the target task datasets. The LUNA16 dataset belongs to the CT domain while the Brain tumor dataset belongs to the MR domain. Similar domain differences can be found in Heart, Hippocampus, and Prostate datasets.

The significant domain difference requires a new Models Genesis trained on the MR domain. However, CT images have a standard scale (Hounsfield Unit), while in MR images, the variation in the scanners causes inconsistent tissue intensities. Due to this reason, the non-linear transformation in Models Genesis does not work well with MR images. Hence, future work is required to find the transformation strategy for images in the MR domain.

## 5.4 Importance of the Normalization Strategy

The nnU-Net framework highly relies on the ground truth while building the pipeline. The normalization strategy only collects the intensity values from foreground region pixels in the raw scans. Based on the intensity values collected, the framework applies z-score normalization on the individual scans. To reduce the use of ground truth, we replaced the intensity clipping and z-score normalization strategy of the nnU-Net framework with the intensity clip in the range of [-325,325], followed by image normalization between [-1,1]. Further, we trained liver and liver tumor segmentation tasks from scratch using the above strategy. Note: this experiment was done as a proof of concept to evaluate the normalization strategy.

| **Normalization** | **Liver** | **Tumor** |
|---|---|---|
| z-score | 96.18 | 63.37 |
| $[-1, 1]$ | **96.37** | **64.41** |

Table 5.1: Liver and Liver Tumor Segmentation. The only change while determining the above results is in the pre-processing step. Intensity clipping and z-score normalization, in the nnU-Net framework, are done based on the intensity values captured from the foreground pixels. [-1,1] is done based on the clip between [-325,325] (no use of ground truth). Scores depicted are Dice scores. Note: The best result is denoted in bold.

As shown in Table 5.1, the standard clipping of [-325,325] followed by the raw scan scaling between [-1,1] does not affect the performance.

## Chapter 6

# CONCLUSION

In this work, we addressed some of the limitations in the nnU-Net framework and the ways to overcome them. First, we proposed the pre-trained model for the nnU-Net framework in order to enhance the performance of the target tasks provided by the Medical Segmentation Decathlon (2019) challenge. This work shows the importance of transfer learning for tasks with limited data. We demonstrated that by fine-tuning Models Genesis based on the architecture determined by the nnU-Net framework, the performance of liver and lung tumor segmentation tasks, indeed, improved significantly. This performance gain is attributed to the scalable, generic, robust image representation learned from the consistent and recurring anatomical structure embedded in medical images. Further, Models Genesis helped in achieving the first rank in liver tumor segmentation task and third rank in lung tumor segmentation task. Due to the outstanding performance of Models Genesis, we plan on using the same architecture utilized by the proxy task for the target tasks with different architecture. Additional experiments in the discussion section suggest utilizing transfer learning within similar architecture and similar domains.

Second, we observed that introducing an advanced segmentation architecture, like UNet++, improved the performance of the liver tumor segmentation task. This eliminates the need for numerous specialized architectures, further eliminates the need of ensembling multiple architectures. In future work, we plan on evaluating UNet++ as a generic architecture to learn the multiple features to span across multiple tasks. Please refer APPENDIX A for codes and pre-trained nnU-Nets.

# REFERENCES

Carreira, J. and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset”, in “proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 6299–6308 (2017).

Chen, L., P. Bentley, K. Mori, K. Misawa, M. Fujiwara and D. Rueckert, “Self-supervised learning for medical image analysis using image context restoration”, Medical image analysis **58**, 101539 (2019a).

Chen, S., K. Ma and Y. Zheng, “Med3d: Transfer learning for 3d medical image analysis”, arXiv preprint arXiv:1904.00625 (2019b).

Çiçek, Ö., 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”, pp. 424–432 (Springer, 2016).

Ciresan, D., 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”, pp. 2843–2851 (2012).

Deng, J., 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”, pp. 248–255 (Ieee, 2009).

Gibson, E., W. Li, C. Sudre, L. Fidon, D. I. Shakir, G. Wang, Z. Eaton-Rosen, R. Gray, T. Doel, Y. Hu *et al.*, “Niftynet: a deep-learning platform for medical imaging”, Computer methods and programs in biomedicine **158**, 113–122 (2018).

Gidaris, S., P. Singh and N. Komodakis, “Unsupervised representation learning by predicting image rotations”, (2018).

He, K., X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition”, (2015).

Isensee, F., 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”, pp. 120–129 (Springer, 2017a).

Isensee, F., 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”, pp. 287–297 (Springer, 2017b).

Isensee, F., 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).

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

Li, X., 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 transactions on medical imaging **37**, 12, 2663–2674 (2018).

Long, J., E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation", (2015).

Medical Segmentation Decathlon, "Medical segmentation decathlon", `http://medicaldecathlon.com/`, Last accessed on 2019 (2019).

Milletari, F., N. Navab and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation", in "2016 fourth international conference on 3D vision (3DV)", pp. 565–571 (IEEE, 2016).

Noroozi, M. and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles", (2017).

Oktay, O., 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).

Pan, S. J. and Q. Yang, "A survey on transfer learning", IEEE Trans. on Knowl. and Data Eng. **22**, 10, 1345–1359, URL `https://doi.org/10.1109/TKDE.2009.191` (2010).

Pathak, D., P. Krahenbuhl, J. Donahue, T. Darrell and A. A. Efros, "Context encoders: Feature learning by inpainting", (2016).

Ronneberger, O., P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in "International Conference on Medical image computing and computer-assisted intervention", pp. 234–241 (Springer, 2015).

Ross, T., D. Zimmerer, A. Vemuri, F. Isensee, M. Wiesenfarth, S. Bodenstedt, F. Both, P. Kessler, M. Wagner, B. Müller *et al.*, "Exploiting the potential of unlabeled endoscopic video data with self-supervised learning", International journal of computer assisted radiology and surgery **13**, 6, 925–933 (2018).

Setio, A. A. A., A. Traverso, T. De Bel, M. S. Berens, C. van den Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts *et al.*, "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge", Medical image analysis **42**, 1–13 (2017).

Simonyan, K. and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", (2015).

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions", (2014).

Tajbakhsh, N., J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?", IEEE transactions on medical imaging **35**, 5, 1299–1312 (2016).

Zhang, R., P. Isola and A. A. Efros, "Colorful image colorization", (2016).

Zhou, Z., M. M. R. Siddiquee, N. Tajbakhsh and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation", in "Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support", pp. 3–11 (Springer, 2018).

Zhou, Z., 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", pp. 384–393 (Springer, 2019).

Zhuang, X., Y. Li, Y. Hu, K. Ma, Y. Yang and Y. Zheng, "Self-supervised feature learning for 3d medical images by playing a rubik's cube", in "International Conference on Medical Image Computing and Computer-Assisted Intervention", pp. 420–428 (Springer, 2019).

# APPENDIX A

# CODE

As open science, all codes and pre-trained nnU-Nets are available at https://github.com/MrGiovanni/ModelsGenesis/tree/master/competition