Integrating Residual, Dense, and Inception Blocks into the nnUNet

Niccolò McConnell
Dept. of Computer Science
Brunel University London
London, UK
niccolo.mcconnell@brunel.ac.uk

Alina Miron
Dept. of Computer Science
Brunel University London
London, UK
alina.miron@brunel.ac.uk

Zidong Wang
Dept. of Computer Science
Brunel University London
London, UK
zidong.wang@brunel.ac.uk

Yongmin Li
Dept. of Computer Science
Brunel University London
London, UK
yongmin.li@brunel.ac.uk

***Abstract*—The nnUNet is a fully automated and generalisable framework which automatically configures the full training pipeline for the segmentation task it is applied on, while taking into account dataset properties and hardware constraints. It utilises a basic UNet type architecture which is self-configuring in terms of topology. In this work, we propose to extend the nnUNet by integrating mechanisms from more advanced UNet variations such as the residual, dense, and inception blocks, resulting in three new nnUNet variations, namely the Residual-nnUNet, Dense-nnUNet, and Inception-nnUNet. We have evaluated the segmentation performance on eight datasets consisting of 20 target anatomical structures. Our results demonstrate that altering network architecture may lead to performance gains, but the extent of gains and the optimally chosen nnUNet variation is dataset dependent.**

***Index Terms*—nnUnet, Biomedical image segmentation, Residual networks, Dense networks, Inception networks.**

I. Introduction

Medical imaging provides a non-invasive tool for clinicians to assess anatomical structures [14]. Within the context of medical imaging, segmentation consists in sub-dividing raw images into distinct anatomical regions, and thereby supports clinicians with evidence during anatomical diagnosis [11], [12]. Manual segmentation is highly time consuming, and subject to intra and inter observer variability, with results dependent on clinicians' experience [4]. Accurate and robust automatic-segmentation methods, therefore, provide the potential for significant impact via increased reproducibility and the improvement of clinical workflows.

The rise of deep learning (DL), and particularly the success of the convolutional neural network, has resulted in DL based methods being the state-of-the-art approach for medical image analysis [4], [10], [11], [18]. The UNet [17] is a fully convolutional encoder-decoder style network which has demonstrated promising results in biomedical image segmentation tasks, with UNet based networks being the most successful DL segmentation approach [18]. DL based segmentation, however, is not solely reliant on network architecture but is also dependent on other components and hyperparameters of the network training pipeline such as dataset preprocessing, image augmentation styles, training batch size etc [10]. The design of the training pipeline is dataset dependent and non-optimal component choices lead to a deterioration in performance.

The nnUNet framework developed by Isensee et al. [9] aimed to address the challenge of optimally selecting training pipeline components without the need for manual trial and error based selection. The nnUNet is a fully automated and generalisable framework which automatically configures the full training pipeline for any segmentation task it is applied on, taking into account dataset properties and hardware constraints. The nnUNet utilises a UNet type architecture which is self-configuring in terms topology (network depth, kernel sizes, and pooling operations); it consists of standard convolutional blocks, and apart from its use of deep supervision, does not utilise architectural features found in the more advanced variations of UNet type architectures such as residual, dense, or inception blocks [9], [18]. The nnUNet achieved state-of-the-art performance on 33 of the 53 anatomical structures, while otherwise attaining results very close to the top of the respective leaderboards [9].

It is emphasised that the impressive performance of nnUNet is not due to an advanced UNet architecture, but instead attributable to utilising a systematic method for tailoring the training pipeline configuration to new datasets [9]. In fact, while the method utilises a relatively standard UNet architecture, it achieved state-of-the-art results on a variety of datasets.

Utilising more advanced architectures, however, may lead to an improvement in specific task performance. The key contribution of this article is the extension of the nnUNet framework via the integration of advanced architectural components, and thereby we propose the following three variations: Residual-nnUNet, Dense-nnUNet, and Inception-nnUNet. The performance of our nnUNet variations is evaluated on 8 medical imaging datasets consisting of 20 anatomical structures, and we demonstrate that optimal network architecture is dataset dependent. Furthermore, we provide source code for the custom variations and thereby allow users to select the nnUNet variant for optimal performance. [1]

II. Related Works

It has been widely reported that the use of advanced UNet architectures attains improvements to segmentation perfor-

[1]Source code of the project is available at https://github.com/niccolo246/Extended_nnUNet.git

mance compared to the standard vanilla UNet [17]. However, nnUnet attains performance at or close to the state-of-the-art in most segmentation tasks using a standard Unet. There is limited research into the integration of more advanced architectures into the nnUNet segmentation framework.

In the original nnUNet paper, Isensee et al. [9] investigated the integration of an alternative UNet inspired network which utilised an asymmetrical fully residual encoder with increased convolutional layers, while the decoder remained unchanged. The standard nnUNet, however, achieved superior average performance on eight of the ten datasets. Isensee et al. [8] later extended nnUNet to be tailored to the 2020 Brain Tumour Segmentation Challenge (BraTS) [2], [13]. The non-architecture related extensions included BraTS specific result post processing, improved augmentation, region based training, increased training batch size, and use of batch Dice. The resulting framework achieved first place in BraTS 2020; the alterations to nnUNet consisted of minor architecture related changes, namely the replacement of instance normalisation layers with batch normalisation layers. Xu et al. [22] proposed an extension to the nnUNet framework which was developed for the 2021 Kidney Tumour Segmentation Challenge (KiTS). The authors adjusted the data augmentation component to be tailored to the challenge, and additionally adopted an ensemble type approach for improved prediction performance. The proposed extension to nnUNet, however, did not involve alterations to the network architecture component of nnUNet. Luu et al. [12] extended nnUNet to be tailored for optimal performance in BraTS 2021, and, in fact, built upon the BraTS 2020 winning submission by Isensee et al. [8] discussed previously. Alterations to the architecture component included the use of an asymmetrically larger UNet encoder, use of axial attention mechanism in the decoder, and replacement of batch normalisation with group normalisation. The Luu et al. nnUNet extension [12] achieved first place on the unseen test data of BraTS 2021.

In this work, we take forward the concept of altering the nnUNet architecture component to create three new nnUNet variations, namely Residual-nnUNet, Dense-nnUNet, and Inception-nnUNet, and evaluate the segmentation performance on multiple datasets.

III. Methods

The prominent advantage of the nnUNet framework lies in its ability to automatically tailor the network training pipeline to any segmentation task on which it is applied, without the need for manual intervention, while attaining results which are in line or superior to expert-configured pipelines. We provide a brief summary of the nnUNet methodology, although for a deep dive into the inner workings, please refer to the original paper by Isensee et al. [9].

Firstly, training data is inputted to nnUNet, and it will automatically retrieve a "data-fingerprint" which consists of information on the data's imaging modality, intensity distribution, median image shape, and median image spacing. Based on the data-fingerprint, heuristic based decisions will be automatically executed, taking into account the system GPU memory constraints, in order to determine the "rule-based parameters", which include image resampling method, label resampling method, training batch size, network topology, patch size to be inputted to network etc. "Fixed parameters" which remain constant regardless of the task being considered include the use of ADAM optimiser, combined Cross Entropy plus Dice loss function, and data augmentation techniques done on the fly during training. The nnUNet also has the ability to train three types of UNet inspired architectures which are the 2D UNet, 3D UNet, and 3D cascade UNet, and based on the results will determine "empirical parameters" which include the result post-processing approach as well as the use of an ensemble method. In this work we concentrate exclusively on the use of the 3D UNet.

A. *Standard-nnUNet*

The Standard-nnUNet utilises a 3D encoder-decoder UNet inspired network in which the encoder and decoder are interlinked with skip connections for improved information preservation during the decoding stage. 3D convolutions are utilised for feature extraction, upsampling is performed via transposed convolutions while downsampling is performed via strided convolutions. The network's convolutional blocks consist of a convolutional layer followed by instance normalisation, and finally a LeakyReLU activation function is applied with gradient 0.01. An example representation of the architecture template is illustrated in Fig. 1a. The nnUNet framework automatically configures network topology via selection of the number of downsampling operations (network depth), kernel sizes, and stride parameters in order to ideally achieve a feature map size of 4x4x4 in the bottleneck layer while using the largest input patch size possible and at least a training batch size of two given hardware constraints. Deep supervision is utilised by default in Standard-nnUNet in order to inject gradients deeper into the network during training; this is achieved by adding auxiliary losses to all but the deepest two layers of the decoder, with the final loss function being a weighted sum of losses at all the relevant depth levels.

A Note on the Architectural Variants: The key nnUNet ability to self-configure network topology is preserved in the more advanced architectures that we subsequently integrated. Therefore, the following visual network representations (as with the standard-UNet mentioned above) are subject to topological changes depending on the dataset utilised and are, hence, meant to serve as a conceptual illustration rather than provide details of actual trained networks.

B. *Residual-nnUNet*

Inspired by the success of ResNet [5], we integrated a fully residual UNet into the architecture component of nnUNet and, hence, propose Residual-nnUNet which incorporates residual connections at depth level as illustrated by the template in Fig 1b. The residual connection performs addition of a convolutional block's input to its output, and thereby enables the network to preserve information from previous layers; the

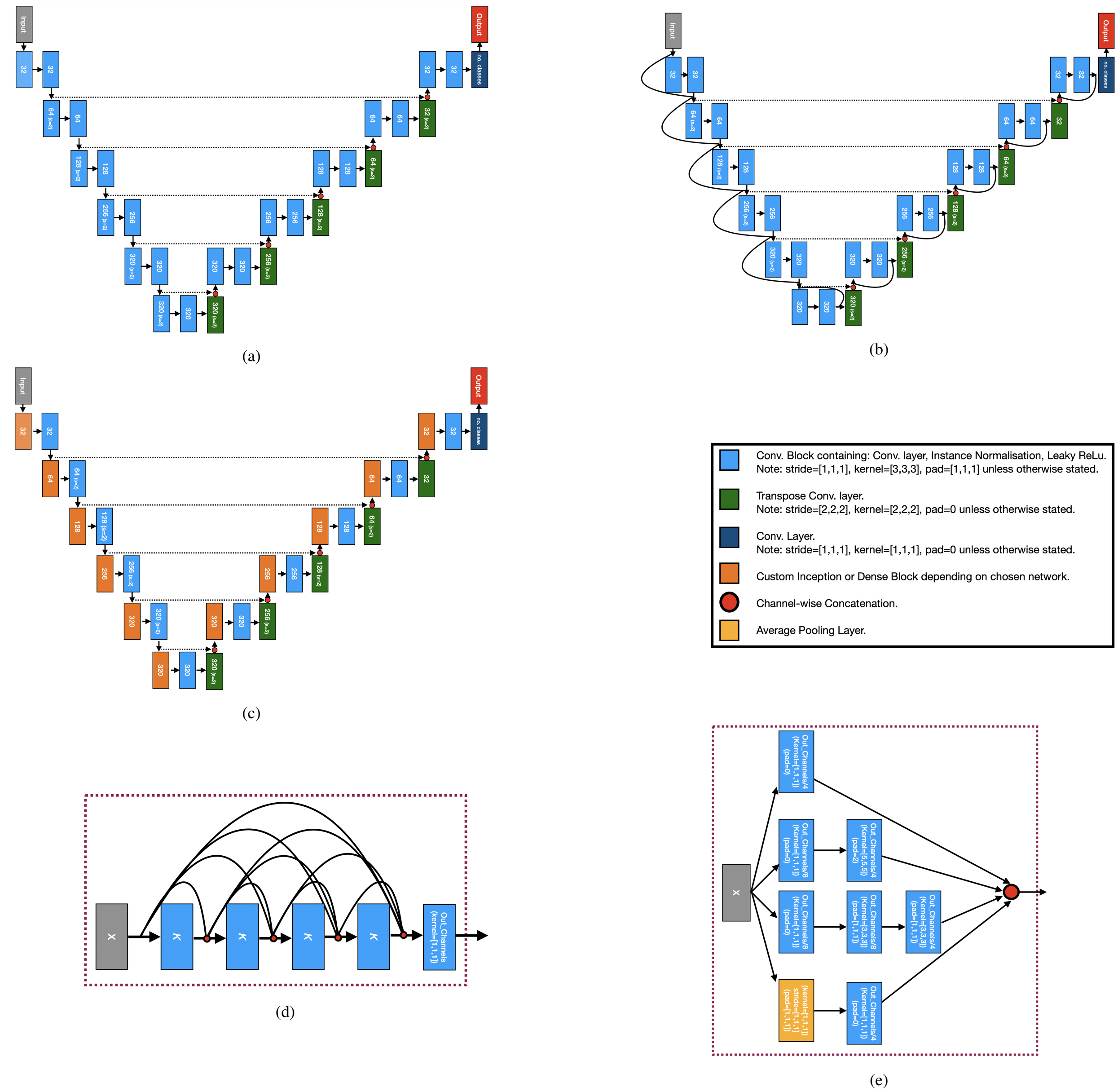

Fig. 1: Graphical architecture representations (quantity on blocks represent output channels). (a) Baseline-UNet representation. (b) Residual-UNet representation. (c) Inception-UNet or Dense-UNet representation, depending on whether orange blocks replaced with Dense block or Inception block, respectively. (d) Dense block representation. (e) Inception block representation. Note: legend is shared across (a)-(e).

residual connections theoretically improve training in deep networks as they ameliorate gradient flow during backpropogation and help mitigate against the vanishing gradient problem [6].

C. Dense-nnUNet

DenseNet [7] can be thought of as an extension of ResNet which instead utilises dense connections. Our proposed DenseNet inspired Dense-nnUNet was designed by integrating dense blocks into nnUNet via the template illustrated in Fig. 1c, in which the orange blocks represent the denseblocks visualised in Fig. 1d.

In a dense block every convolutional sub-block (consisting of convolutional layer, instance normalisation, LeakyReLU) receives as input the channel-wise concatenated feature maps of all previous sub-blocks, with each sub-block outputting K channels (K represents the growth rate). The dense connections allow for deeper networks through improved preservation of information between layers, improved gradient flow, and implicit deep supervision [7].

Our defined dense block consists in four convolutional sub-blocks, with each sub-block having growth rate K=10. A key element for nnUNet integration is the dense block's use of a 1x1x1 kernel in its final convolutional layer, as this allows the dense block to output the nnUNet-determined number of channels (thereby preserving its self-configuring nature) to be passed on to the rest of the network.

D. *Inception-nnUNet*

Inception UNet variants are inspired by Google's Inception-Net [20], with inception blocks finding success in being integrated into vanilla UNet architectures [3]. The idea being instead of choosing a fixed convolutional filter size for each layer, one can use an inception block in which multiple filter sizes are present, with the resulting output from each filter then being concatenated together. Inspired by Szegedy et al. [20], we defined the inception block illustrated in Fig. 1e, in which the input is passed to four branches which include kernel size 1x1x1, kernel size 3x3x3, kernel size 5x5x5 and an average pool operation. The outputs of each branch then undergo channel-wise concatenation with the block's final output having the nnUNet-determined number of channels to be passed on to the rest of the network (preserving its self-configuring nature). The correct final output channels is achieved by having each branch's penultimate layer output a quarter of the desired number of final output channels as there are four total branches (although this could be optionally altered by including use of the 1x1x1 convolutional layer as was done with dense block in Fig. 1d).

Our proposed Inception-nnUNet was defined by integrating our custom inception block into nnUNet via the template illustrated in Fig. 1c, in which the orange blocks represent the aforementioned inception blocks from Fig. 1e.

IV. Results

A brief summary of the eight explored datasets is presented in table I. Datasets D1-D7 originate from the Medical Segmentation Decathlon Challenge [1], [19] whereas dataset D8 originates from the 2021 Fetal Brain Tissue Annotation and Segmentation Challenge (FeTA) [16] - for a more in depth description of the datasets, refer to the respective challenge papers. As there are no ground-truth labels in the testing sets, we used only the training sets in our experiments. Because all the training, validation, and testing splits happen locally for each dataset, this results in some tasks having a reasonably restricted training dataset which may negatively impact performance.

We trained five different nnUNet variants for datasets D1-D8 on an Nvidia A6000 GPU. The average Dice scores for each anatomical region are presented in Fig. 2; Dice score was selected as the evaluation metric as it is the most commonly utilised metric for the validation of medical volume segmentations [21]. Standard-nnUNet refers to the original nnUNet discussed in section III-A which makes use of deep supervision, while Baseline-nnUNet is equivalent to Standard-nnUNet although without the use of deep supervision. Residual-nnUNet, Dense-nnUNet, and Inception-nnUNet refer to our proposed nnUNet extensions which make use of residual (section III-B), dense (section III-C), and inception (section III-D) blocks, respectively; note that the deep supervision feature is preserved throughout our proposed nnUNet variations.

V. Discussion

It is observed that different nnUNet variants performed optimally on different respective datasets. Overall, the Baseline-nnUNet (deep supervision free) failed to achieve best performance on any of the datasets, relative to the other four architectures, while the Standard-nnUNet achieved best performance on at least one anatomical region in five of the eight datasets. The Residual-nnUNet, Dense-nnUNet, and Inception-nnUNet achieved top performance on at least one anatomical region in five, three, and two of the explored datasets, respectively.

Relative to Standard-nnUNet, the Baseline-nnUNet displayed a drop in average performance ranging up to -7.98% and attained lower performance on 17 of the 20 explored anatomical structures, with the model never achieving top performance on any datasets. We therefore hypothesise that the removal of deep supervision generally resulted in decreased performance due to reduced gradient flow, which is especially relevant for deeper UNet architectures. This may also explain why the nnUNet variations which make use of residual (Residual-nnUNet) and dense (Dense-nnUNet) features generally resulted in marginal gains in performance compared to the Standard-nnUNet as their theoretical aim is to improve gradient flow which is already helped via the incorporation of deep supervision. In fact, our Residual-nnUNet and Dense-nnUNet variants attained a performance improvement of up to 2.99% and 15.02%, respectively, compared to Standard-nnUNet. This compares to a performance improvement of up to 6.74% and 25.00%, respectively, when Residual-nnUNet and Dense-nnUNet variants are, respectively, compared to Baseline-nnUNet. The results, hence, suggest that the deep supervision is indeed an important feature for performance, and is, therefore, appropriately utilised in the original nnUNet framework by default.

Overall, the results suggest that altering the nnUNet network architecture has a marginal effect on performance across the majority of the datasets explored, which was in line with our expectations as the nnUNet framework currently performs at or close to state-of-the-art, with the differences in performance for top ranked competition entries also being marginal. Importantly, however, the best performing nnUNet variation as well as the extent of performance gains is dependant on the dataset in question, as was evidenced by the use of Dense-nnUNet which increased average Dice score for the D5 Tumour region by 25.00% compared to Baseline-nnUNet. While the consistent use any one single architecture does not appear to significantly impact the average performance and generalisability of the nnUNet framework across several datasets, the optimal selection of an architecture depending on the specific dataset in question may result in considerable or marginal gains in performance which could be sufficient to establish a new state-of-the-art result. Consequently, for users who wish to maximise performance on a specific dataset, it is worth experimenting with alternative network architectures integrated into nnUNet such as the ones we proposed.

Dataset	Modality	Regions of Interest	Median Volume Size (Voxel)	Median Volume Spacing (mm)	No. Cases (train/valid/test)
D1 - Brain	Multi-modal MRI (FLAIR, T1w, T1w_Gd, T2w)	Edema, non-enhancing tumour, enhancing tumour	[138, 169, 138]	[1.00, 1.00, 1.00]	484 (290/ 73 / 121)
D2 - Hippocampus	Mono-modal MRI	Anterior hippocampus, posterior hippocampus	[40, 56, 40]	[1.00, 1.00, 1.00]	260 (166/ 42 / 52)
D3 - Liver	Portal venous phase CT	Liver, liver tumour	[482, 512, 512]	[1.00, 0.76, 0.76]	129 (68/ 18 / 43)
D4 - Lung	CT	Lung, lung cancer	[253, 512, 512]	[1.25, 0.79, 0.79]	63 (33/ 9 / 21)
D5 - Pancreas	Portal venous phase CT	Pancreas, pancreatic tumour mass	[96, 512, 512]	[2.50, 0.79, 0.79]	280 (179/ 45 / 56)
D6 - Colon	CT	Colon cancer primaries	[152, 512, 512]	[3.00, 0.78, 0.78]	126 (67/ 17 / 42)
D7 - Hepatic Vessels	CT	Hepatic vessels, hepatic tumour	[150, 512, 512]	[1.50, 0.80, 0.80]	303 (161/ 41 / 101)
D8 - Fetal Brain	Mono-modal MRI	External cerebrospinal fluid, grey matter, white matter, ventricles, cerebellum, deep grey matter, brainstem/ spinal-cord	[256, 256, 256]	[0.50, 0.50, 0.50]	80 (51/ 13/ 16)

TABLE I: Summary of explored datasets. MRI—magnetic resonance imaging, FLAIR—fluid-attenuated inversion recovery, T1w—T1 weighted image, T1w_Gd—post-Gadolinium (Gd) contrast T1-weighted image, T2w—T2 weighted image, CT—computed [19].

Finally, we hypothesise that further potentially significant gains in performance may be attained by experimenting with altering the architecture in tandem with altering other elements of the nnUNet training pipeline, although these are likely to be dataset specific as mentioned section II. A current limitation of this work is the restricted train/validation/test dataset utilised. Our results are therefore not directly comparable to the original dataset competition leaderboards as we use a subset of the train dataset as our test dataset seeing as the official competition test dataset do not have open access labels. A further limitation is due to each volume in the explored datasets consisting of only one ground-truth segmentation map, and therefore we are unable to determine the expert-performance as well as inter-observer variability for each task.

Future avenues of exploration could include investigating performance from combining the different network components explored in this work, as well as inclusion of the UNet attention mechanism [15]. Furthermore, we plan to explore potential performance gains resulting from the ensembling of more advanced UNet network variants.

VI. Conclusion

In this work, we have presented three extensions to the nnUNet framework, namely the Residual-nnUNet, Dense-nnUNet, and Inception-nnUNet, inspired from the ResNet [5], DenseNet [7] and Inception-Net [20], respectively. Experiments on eight medical imaging datasets consisting of 20 anatomical structures demonstrate that the deep supervision plays an important role in all nnUNet variants (including the original Standard-nnUNet) and especially for deeper UNet architectures, with the Baseline-nnUNet without deep supervision not attaining top performance on any of the datasets. Among the variants with deep supervision, while altering network architecture may result in performance gains, both the extent of the performance increase and the optimal network variation are dataset dependent. We have made the source code publicly available which allows future users to easily experiment with multiple nnUNet variations and select the modification which provides optimal performance on their specific dataset.

References

[1] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, Bram van Ginneken, et al. The medical segmentation decathlon. *arXiv preprint arXiv:2106.05735*, 2021.

[2] Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, Sung Min Ha, Martin 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.

[3] Liang Chen, Paul Bentley, Kensaku Mori, Kazunari Misawa, Michitaka Fujiwara, and Daniel Rueckert. Drinet for medical image segmentation. *IEEE Transactions on Medical Imaging*, PP:1–1, 05 2018.

[4] Intisar Rizwan I Haque and Jeremiah Neubert. Deep learning approaches to biomedical image segmentation. *Informatics in Medicine Unlocked*, 18:100297, 2020.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6:107–116, 04 1998.

[7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[8] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

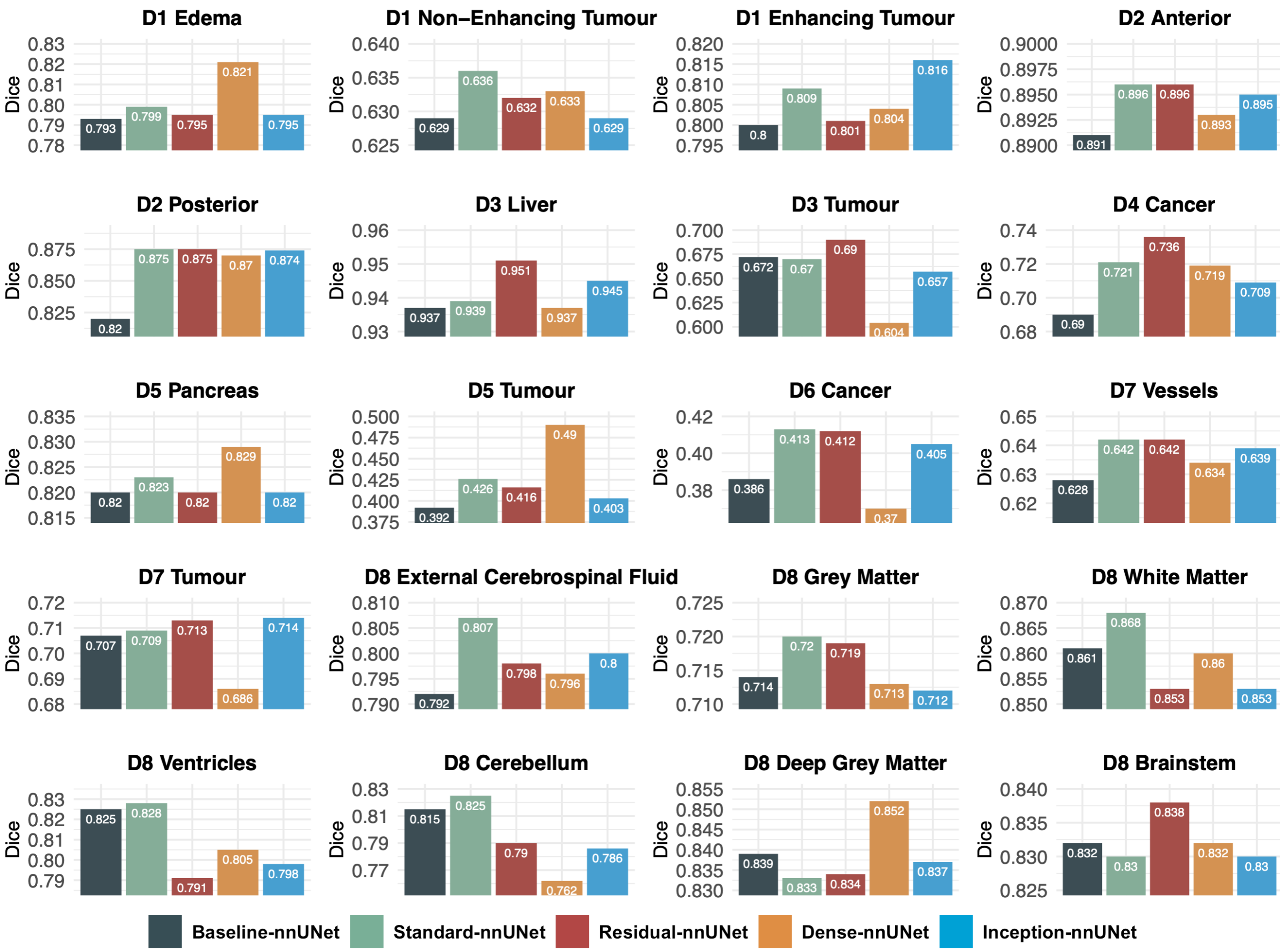

Fig. 2: Graph showing average Dice scores attained by each nnUNet variation for the anatomical regions from datasets D1 - D8.

[9] Fabian Isensee, Paul F Jäger, Peter M Full, Philipp Vollmuth, and Klaus H Maier-Hein. nnu-net for brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 118–132. Springer, 2020.

[10] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[11] Liangliang Liu, Jianhong Cheng, Quan Quan, Fang-Xiang Wu, Yu-Ping Wang, and Jianxin Wang. A survey on u-shaped networks in medical image segmentations. *Neurocomputing*, 409:244–258, 2020.

[12] Huan Minh Luu and Sung-Hong Park. Extending nn-unet for brain tumor segmentation. *arXiv preprint arXiv:2112.04653*, 2021.

[13] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[14] European Society of Radiology (ESR) communications@ myesr. org. Medical imaging in personalised medicine: a white paper of the research committee of the european society of radiology (esr). *Insights into imaging*, 6:141–155, 2015.

[15] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[16] Kelly Payette, Priscille de Dumast, Hamza Kebiri, Ivan Ezhov, Johannes C Paetzold, Suprosanna Shit, Asim Iqbal, Romesa Khan, Raimund Kottke, Patrice Grehten, et al. An automatic multi-tissue human fetal brain segmentation benchmark using the fetal tissue annotation dataset. *Scientific Data*, 8(1):1–14, 2021.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[18] Nahian Siddique, Sidike Paheding, Colin P Elkin, and Vijay Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access*, 2021.

[19] Amber L Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram Van Ginneken, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*, 2019.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[21] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging*, 15, 08 2015.

[22] Lizhan Xu, Jiacheng Shi, and Zhangfu Dong. Modified nnu-net for the miccai kits21 challenge. In *International Challenge on Kidney and Kidney Tumor Segmentation*, pages 22–27. Springer, 2022.