



# Autofocus Net: Auto-focused 3D CNN for Brain Tumour Segmentation

Andreas Stefani<sup>1</sup>, Roushanak Rahmat<sup>2</sup>, and David Harris-Birtill<sup>1</sup>

<sup>1</sup> School of Computer Science, University of St Andrews, St Andrews, UK  
[research@andreasstefani.de](mailto:research@andreasstefani.de)

<sup>2</sup> The Institute of Cancer Research, Sutton, UK

**Abstract.** Several approaches based on convolutional neural networks (CNNs) are only able to process 2D images while most brain data consists of 3D volumes. Recent network architectures which have demonstrated promising results are able to process 3D images. In this work, we propose an adapted approach based on a CNN to process 3D contextual information in brain MRI scans for the challenging task of brain tumour segmentation. Our CNN is trained end-to-end on multi-modal MRI volumes and is able to predict segmentation for the binary case, which segments the whole tumour, and multi-class case, which segments the whole tumour (WT), tumour core (TC) and enhancing tumour (ET). Our network includes multiple layers of dilated convolutions and autofocus convolutions with residual connections to improve segmentation performance. Autofocus layers consist of multiple parallel convolutions each with a different dilation rate. We replaced standard convolutional layers with autofocus layers to adaptively change the size of the effective receptive field to generate more powerful features. Experiments with our autofocus settings on the BraTS 2018 glioma dataset show that the proposed method achieved average Dice scores of 83.92 for WT in the binary case and 66.88, 55.16, 64.13 for WT, TC and ET, respectively, in the multi-class case. We introduce the first publicly and freely available NiftyNet-based implementation of the autofocus convolutional layer for semantic image segmentation.

**Keywords:** Deep learning · 3D CNN · MRI · Brain tumour segmentation · Autofocus layer · Glioma

## 1 Introduction

In the UK, the National Health Service (NHS) [16] reports over 11,000 patients yearly diagnosed with primary brain tumours originating in the brain with half of those diagnosed with a cancerous tumour. Beyond that many others are diagnosed with secondary brain tumours, whose cells have broken away from the primary tumour residing in another part of the body and have spread to the brain. Glioblastoma (GBM) is the most common and aggressive primary brain tumour in adults [14]. In particular, GBM is one group of the type Glioma that

starts in the glial cells, which can be categorised into low-grade gliomas (LGG), which are benign and slow growing, or high-grade gliomas (HGG), which are malignant, fast growing and more aggressive with worse prognosis [4, 25]. To improve the survival rate of patients, brain tumour segmentation is an important task during diagnosis, treatment planning and treatment itself: The location and size of the tumour needs to be determined, the tumour boundary precisely outlined to expose very little healthy tissue to radiation, and the treatment progress evaluated and monitored [2, 26].

To detect gliomas, experts can use magnetic resonance imaging (MRI) to create images of the brain [5]. Usually the generated images are in 2D, putting stacks of 2D slices together produces a 3D model of the brain. Varying settings in MRI can produce different MRI sequences, resulting in different image appearance as shown in Fig. 3. Using different sequences for segmenting brain tumours lead to better results since the modalities provide complementary information for different glioma sub-regions [13].

Compared to manual segmentation, automatic brain tumour segmentation has proven the potential for more accurate and reliable results to aid precise treatment of brain tumours. Here the task is to segment brain MRI scans to highlight potential tumours with the aim to determine the outline of tumour structures as accurately as possible. Various machine learning approaches to medical image segmentation emerged over the last decades [12]. Significant progress has been achieved in recent years due to the advent of convolutional neural networks (CNNs) which have outperformed previous state-of-the-art methods [10, 12, 22].

A vast majority of end-to-end trained 2D and 3D CNN architectures have been proposed to tackle the task of medical image segmentation. According to Litjens et al. [12], most of the brain images are 3D volumes. Therefore, using a 2D network on 3D image volumes might be suboptimal due to unused image information along the missing third axis, which served as an inspiration of our work. In [21] the 2D U-Net which processes 2D slices for biomedical image analysis was proposed. Based on this, the 3D U-Net in [3] is capable of processing 3D image volumes by replacing all 2D operations with its 3D counterparts. In [8], DeepMedic, a dual-pathway 3D CNN combined with a fully connected 3D conditional random field (CRF) in a post-processing step for automatic brain lesion segmentation was proposed. By processing 3D MRI volumes, they make better utilization of 3D contextual information compared to the methods of [5, 17, 27], which process 2D slices. In [7] the DeepMedic model in [8] was extended with residual connections to achieve better segmentation results for all classes of the segmentation task. V-Net [15] is also a fully convolutional neural network for volumetric image segmentation and allows for processing 3D MRI volumes. In [11], HighRes3dNet for segmentation of fine structures in volumetric images was proposed. The network incorporates high spatial resolution feature maps throughout layers, which is different compared to the downsample-upsample [3, 8, 15] approach, where low-level features with high spatial resolutions will be downsampled at first and resulting feature maps will be upsampled to achieve high-resolution segmentation. The key idea of the architecture is to use dilated convolutions,

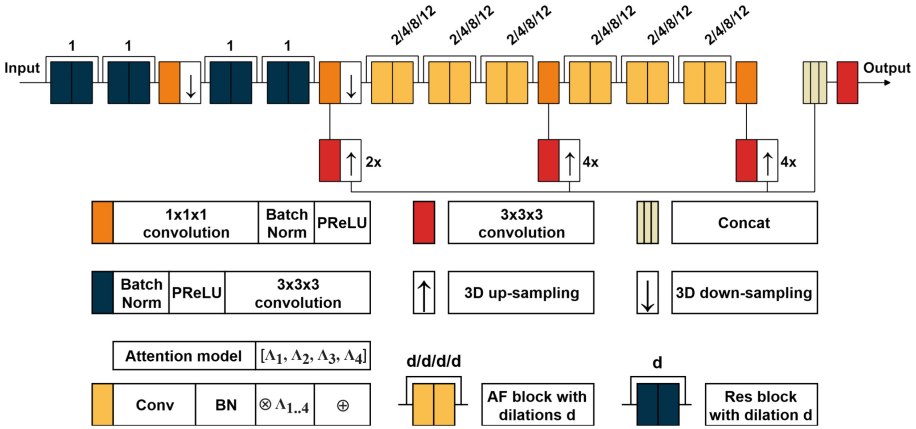
which can be used to produce accurate dense predictions and detailed segmentation maps along object boundaries, and residual connections, which make information propagation smooth and improve the training speed. In [23], a cascade of three 2D CNNs (WNet, TNet, and ENet) was applied on three orthogonal views (axial, coronal, and sagittal) to segment brain tumour sub-regions hierarchically and sequentially from 3D MRI volumes. The segmentation outputs in the three orthogonal views will be fused for a more robust segmentation performance. Instead of using the WNet model for merely predicting the whole tumour as in [23], a WNet was trained in [24] for multi-class segmentation to predict all tumour sub-regions. Investigating CNN enhancements, there has been an approach based on introducing the autofocus layer, which adaptively change the size of the effective receptive field, for semantic image segmentation with CNNs [18]. Models such as [7] were augmented with autofocus layers for the task of brain tumour segmentation to demonstrate promising segmentation results.

In this work we present our end-to-end trained approach to brain tumour segmentation that leverages the capability of CNNs with autofocus layers to process 3D MRI volumes. The network is based on the WNet architecture, which consists of anisotropic, dilated convolutions, and up-/down-sampling to process stacks of 2D volumes [23]. Since most networks process 3D contextual information given 3D MRI volumes, we also propose to process them with 3D operations by adapting the WNet accordingly, in particular, all 2D operations were adapted to its 3D counterparts. Moreover, our proposed Autofocus Net comprises of autofocus layers based on the potential for generating more powerful features by adaptively changing the size of the effective receptive field. This was done by changing certain standard convolutional layers to autofocus layers. We provide a comparison (see Sect. 4) with models without autofocus layers which were evaluated on the same data. Compared with the original 2D WNet model from Wang et al. our resulting 3D model does not match their dice score. For whole tumour segmentation, their binary 2D WNet achieves a dice score of 89.97 whereas our proposed 3D models achieve 79.78 and 83.92 with autofocus layer settings  $\{1, 2, 3\}$  and  $\{2, 4, 8, 12\}$ , respectively.

## 2 Autofocus Net

In Fig. 1, we provide a schematic representation of our CNN. Our network is made up of a modified version of WNet [23] and autofocus (AF) layers [18].

Following the example of 3D U-Net, which is an adapted version of the originally proposed 2D U-Net as mentioned in the introduction, we adapted the 2D WNet accordingly to create a 3D WNet in the first step. Furthermore, considering the convolutions in an autofocus layer [18], which also use 3D shaped kernels with size  $3 \times 3 \times 3$ , it was a necessary step to adapt the WNet accordingly to be able to experiment with autofocus layers. All decomposed kernels of size  $3 \times 3 \times 1$  (intra-slice) and  $1 \times 1 \times 3$  (inter-slice) in WNet were changed into 3D shaped kernels with size  $3 \times 3 \times 3$  and  $1 \times 1 \times 1$ , respectively. Originally, WNet use 10 residual blocks consisting of two convolutional layers with a residual connection bypassing the parameterized layers in the network. Instead, only the first



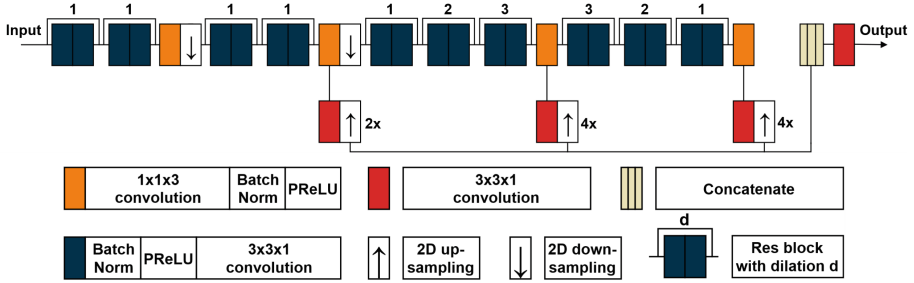
**Fig. 1.** Schematic representation of our CNN architecture. Our network with autofocus layers and dilation rates  $\{2, 4, 8, 12\}$ . The last 6 blocks are autofocus blocks, consisting of two autofocus layers and residual connection as the default residual blocks. The numbers ( $2\times$  and  $4\times$ ) denote twofold and fourfold up-sampling, respectively.

four residual blocks were kept and the last six residual blocks converted into AF blocks, each consisting of two autofocus layers with a residual connection.

Two Autofocus Net models were trained: one with dilation rates  $\{1, 2, 3\}$  and another one with dilation rates  $\{2, 4, 8, 12\}$ . Both were trained for the binary and multi-class case with all hyper-parameters adopted from the original configuration in [23] with a modification in batch size due to hardware limitations in our setup. The models were trained in axial, coronal and sagittal view and the results were fused (see Sect. 3.3) for evaluation. The Adaptive Moment Estimation (Adam) [9] optimiser was used for training, with initial learning rate  $1e-3$ , weight decay  $1e-7$ , regularisation type L2 and batch size 1. The size of the training image windows sampled from the image volumes was  $96 \times 96 \times 96$  for the modalities (T1, T2, T1c, FLAIR) and the labels (Seg). Due to class imbalanced data, the Dice loss function [15] was used for training the binary Autofocus Net. For the multi-class case, a combination of Dice and Cross-Entropy loss used in [6] was used for highly unbalanced segmentations.

## 2.1 WNet

The WNet network of the three cascaded networks from [23] has been implemented in NiftyNet [19]. In [23], the WNet is merely used for binary segmentation to segment the whole tumour (WT) in the first step before passing the segmentation result on to the second network in the sequence for further processing. However, in [24] a single WNet was trained for multi-class segmentation to segment all tumour sub-regions at once. This indicates the possibility to train the WNet for binary and multi-class segmentation (Fig. 2).



**Fig. 2.** WNet: Convolutional layers (blue) have a different ordering of the components (Batch Norm, Activation and Convolution) compared to the architecture in [23]. Numbers (2× and 4×) denote twofold and fourfold 2D up-sampling, respectively (Adapted from [23]). (Color figure online)

The architecture consists of 10 residual blocks with anisotropic convolutions, dilated convolutions, and multi-scale prediction to improve segmentation performance. Each of the 20 convolutional layers in a residual block is an intra-slice layer with kernel size  $3 \times 3 \times 1$ . The four inter-slice convolutional layers have a kernel size of  $1 \times 1 \times 3$ . This decomposition of a 3D kernel ( $3 \times 3 \times 3$ ) into an intra-slice kernel and inter-slice kernel deals with anisotropic receptive fields. The 2D up-sampling layers make sure that all dimensions of the tensors are an equal size when applying the concatenate operation.

NiftyNet provides a pre-trained WNet model trained on the BraTS 2017 dataset, which we could have used as a sensible starting point for transfer learning to utilise the learning from the model. However, it was not possible to ensure that our test data is not in their training data to avoid biased results. Therefore, we retrained the WNet with our own training data from the BraTS 2018 dataset for binary and multi-class segmentation as a baseline for comparison with our Autofocus Net. In both cases, the WNet was trained in axial, coronal and sagittal view and the 3D results were fused as described in Sect. 3.3 to get an average segmentation result which reduces false positives. For comparison we also compared the results of our model with the pre-trained model on our test data to make inferences with the knowledge that the pre-trained model might be biased given that some of our test cases could have been in their training data due to overlapping HGG and LGG volumes in BraTS 2017 and BraTS 2018 dataset.

## 2.2 Autofocus Layer

The autofocus layer can be integrated into existing network architectures by replacing standard or dilated convolutions [18]. These layers adaptively change the size of the effective receptive field to generate more powerful features. In particular, autofocus layers enhance the multi-scale processing of CNNs by adaptively choosing the optimal scale for identifying different objects in an image. Additionally, it enhances the interpretability of a CNN as the computed

attention maps, which are used as filters, unveiling how the zoom at different scales will be realised.

An autofocus layer, as depicted in [18], consists of an attention model and  $K$  (e.g.  $K = 4$ ) parallel dilated convolutions each with a different dilation rate ( $r^k$ ) to capture multi-scale information. Given the activations of the previous layer  $F_{l-1}$ , each dilated convolution processes the same input  $F_{l-1}$ , where  $l$  denotes the layer number. Therefore, it is necessary to create  $K$  convolutions each with a different dilation rate. Note that all parallel convolutions must share the same weights to be adaptively scale invariant. However, there is no possibility in NiftyNet to share weights between multiple convolutions.

Our approach was to use the same convolutional layer multiple times, each time with another input. This simulates the desired parallelism and solves the weight sharing issue. However, using this approach does not solve the problem of having different dilation rates in each parallel layer. There is merely one convolutional layer for weight sharing as mentioned before with a certain dilation rate at a time. Hence, instead of creating  $K$  parallel convolutions with different dilation rates,  $K$  dilated input tensors each with a different dilation rate were created. Using those  $K$  tensors as input for a single convolution (dilation rate = 1) one after another simulates the desired parallelism and has the same effect.

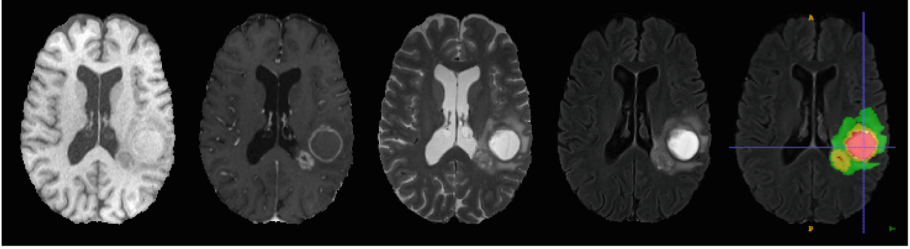
## 3 Implementation Details

### 3.1 Data

We used MRI scans from 285 glioma patients from the 2018 BraTS Challenge [1, 14, 20] dataset. The training set consists of 210 HGG and 75 LGG cases. For each case, it includes 4 modalities (T1, T1c, T2, and FLAIR) that were co-registered and resampled to  $1 \text{ mm}^3$  isotropic resolution. In addition, segmentation files with pixel-wise ground truths (GT) that annotate the complete tumour are provided in the training set. For the validation set, it consists of 67 cases each with 4 modalities as in the training set. The GT segmentations of the validation cases are hidden and not provided within the dataset, which is why a random 70/20/10 (in %) split was generated out of the training set to get 199 training (145 HGG, 54 LGG), 57 validation (44 HGG, 13 LGG), and 29 inference (21 HGG, 8 LGG) cases, respectively. The matrix size for each patient is 240 (slice width)  $\times$  240 (slice height)  $\times$  155 (number of slices)  $\times$  4 (modalities).

### 3.2 Training

All images processed by the network are from varying volume size due to cropping all images with a bounding box to remove unnecessary background, which is all voxels with intensity value zero. For example, a volume with dimensions  $240 \times 240 \times 155$  might result in  $134 \times 167 \times 135$ . However, this has no effect other than reducing the image volume sizes across all images. Minimal pre-processing of normalizing the brain-tissue intensities of each sequence to have zero-mean and



**Fig. 3.** MRI modalities. From left to right: T1, T1c, T2, FLAIR, and the ground truth segmentation. Tumour sub-regions: green—edema, yellow—enhancing tumour, and red—necrotic and non-enhancing tumour core. (Color figure online)

unit variance was performed. All networks were trained without data augmentation to ensure more comparability between network architectures. All models were trained for 30,000 iterations and saved every 5,000 iterations which leads to 6 model-checkpoints. The model-checkpoint with the best Dice score was chosen in each case and presented here. Training and evaluation was implemented on a standard workstation with NVIDIA GEFORCE GTX 1060 6 GB GPU. We used Python with the NiftyNet [19] library for implementing the proposed Autofocus Net<sup>3</sup> and the autofocus layer<sup>1</sup>.

### 3.3 Testing

To evaluate model performance, inferences were made on the 29 inference cases generated in the data split. A previously unseen MRI volume can be segmented by processing sampled image windows from an image through the network. The softmax output of the final classification layer of a network consists of a segmentation map, in which each voxel has been assigned a probability between 0 and 1 whether or not belonging to the corresponding tumour sub-region. These probabilities were used to calculate the reported measurements.

The respective 3D predictions for axial, coronal and sagittal view were post-processed. As in [23], fusing the three single predictions was conducted to get an average segmentation result of the three softmax outputs of all views. The output is an array from size  $H \times W \times D \times 1 \times C$  with  $C$  components; one component for each class. With two components (background and WT) in the binary cases and four components (background, WT, ET and TC) in the multi-class cases. When calculating the Dice across segmentation results and channel by channel (WT, ET and TC), all voxels with probability  $>0.5$  are considered as belonging to the corresponding foreground class depending on the current channel. Every other voxel is interpreted as belonging to the background and not part of the class.

<sup>1</sup> Available at <https://github.com/andreasstefani/autofocus-net>.

## 4 Results

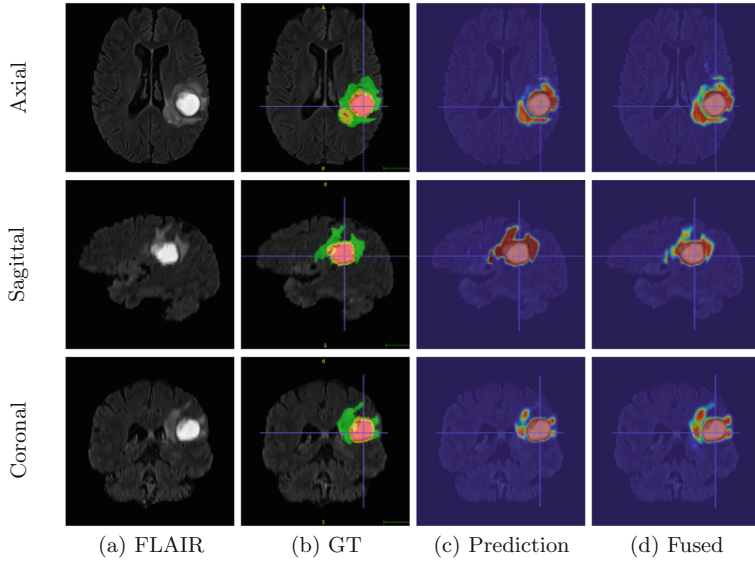
For simplicity of visualisation, only slices of softmax segmentation results for Autofocus Net are shown. Table 1 shows quantitative comparison of the best mean Dice scores with standard deviation (Std) across all trained binary and multi-class models. For multi-class cases, some models have a higher mean Dice for WT but lower mean Dice for TC and ET compared to others. Therefore, the multi-class models with the highest sum of mean Dice scores (WT + TC + ET) were chosen.

**Table 1.** Quantitative comparison between the proposed approach and related networks on the BraTS 2018 dataset.

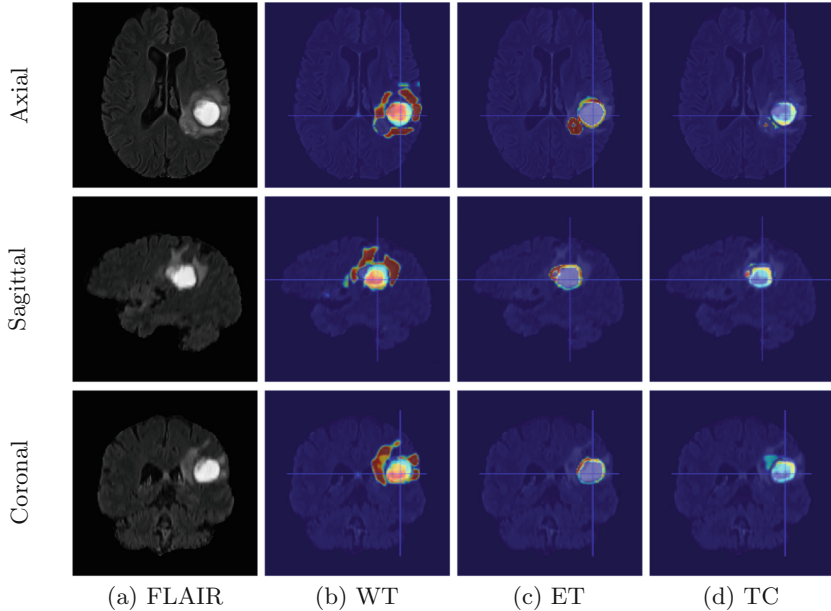
Model	Dice(Std)		
	WT	TC	ET
Pre-trained 2D WNet	89.97(6.50)	×	×
Binary 2D WNet	83.62(13.11)	×	×
Binary 3D WNet	85.72(10.89)	×	×
Binary 3D Autofocus Net{1,2,3}	79.78(12.41)	×	×
Binary 3D Autofocus Net{2,4,8,12}	83.92(11.92)	×	×
Multi 2D WNet	69.82(19.38)	54.93(27.75)	63.06(31.47)
Multi 3D WNet	68.10(18.68)	57.07(26.68)	62.97(29.40)
Multi 3D Autofocus Net{1,2,3}	68.27(19.42)	53.39(26.99)	62.42(31.21)
Multi 3D Autofocus Net{2,4,8,12}	66.88(20.97)	55.16(27.96)	64.13(29.00)

Qualitative results from one inference case using our Autofocus Net for binary and multi-class segmentation, are shown in Fig. 4 and Fig. 5 respectively. The green colour in Fig. 4 (b) shows the segmentation of the whole tumour and represents the edema (label 2). Segmentation images used in Fig. 4 (c) are taken from the corresponding model trained in axial, sagittal or coronal view, respectively. The fused result, combining the three 3D segmentation results of each model is shown in Fig. 4 (d). The fused result per tumour sub-region and view is shown in Fig. 5. In both, a probability is given on a scale between 0 (blue) and 1 (red) whether or not belonging to the corresponding tumour sub-region. Iterations, which achieved best Dice scores are 2D WNet (20k/20k), 3D WNet (30k/30k), Autofocus Net {1, 2, 3} (20k/25k), and Autofocus Net {2, 4, 8, 12} (20k/30k) for (binary/multi-class) models, respectively.





**Fig. 4.** Binary result: Softmax output, which assigns a probability to each voxel. (a) FLAIR. (b) GT - ground truth. (c) predicted binary segmentation. (d) fused prediction. (Color figure online)



**Fig. 5.** Multi-class result: Softmax output, which assigns a probability to each voxel. (a) FLAIR. (b) predicted whole tumour (WT). (c) predicted enhanced tumour (ET). (d) predicted tumour core (TC). (Color figure online)

## 5 Discussion

Our initial assumption that architectural modifications by replacing standard convolutional layers with autofocus layers can boost the segmentation performance proved incorrect as the results show otherwise. In both cases, binary and multi-class, the modified networks have relatively similar or worse dice score results compared to the corresponding model without autofocus layers. However, the networks using autofocus layers with four parallel convolutions perform better than autofocus layers with three parallel convolutions. This suggests an increased number of parallel convolutions enables the autofocus layer to handle the variability of brain tumour sizes more effectively. One drawback of the Autofocus Net is its increased number of additional (parallel) convolutional layers, and therefore a higher number of parameters, inside the autofocus layers that requires longer time for training compared with its non-autofocus counterpart. Regardless if its a binary or multi-class model, training the Autofocus Net using four parallel convolutions inside autofocus layers takes about 12.1% and 13.6% more time for the binary and multi-class model, respectively. For the binary model, our tests showed it took 117,179 s for with autofocus layers and 103,142 s without, and for the multi-class model, it took 124,837 s and 111,383 s for with autofocus layers and without, respectively. However, we believe this is not an important issue considering other networks with similar structure complexity.

The results show that our proposed approach achieved competitive performance for automatic brain tumour segmentation in the binary case. Our multi-class network results differ greatly (up to 20%) from the results in [24] even with a similar network structure and training hyper-parameters. We believe there are two major reasons for this. First, we used a random data split of the training set and thus had 86 MRI volumes (approx. 30%) less for training. Second, the used training hyper-parameters adopted from simpler network structures, in particular the number of training iterations, might be non-optimal for our model complexity. Given a larger time-scale, one could investigate the proper number of autofocus layers and settings (i.e. number of parallel convolutions and dilation rates) for the model with the best training settings to identify the best model for achieving better segmentation results. Additionally, NiftyNet comprises several degrees of freedom regarding pre-processing, such as normalisation, and training, such as data augmentation, optimizer settings or loss function. Choices in both steps are not independent from each other and impact the reproducibility of model performance in the literature heavily and thus an accurate segmentation result which can also be improved in the future.

## 6 Conclusion

In conclusion, we presented an approach based on a 3D CNN with autofocus layers that performs binary and multi-class segmentation of 3D brain MR images. We developed the first publicly available NiftyNet-based implementation of the autofocus convolutional layer [18] that was used in the proposed network.

The approach in this paper was based on the methods proposed in [18,23,24], as we took advantage of them. We used the WNet and adapted it from 2D to 3D and additionally replaced a part of the last standard convolutional layers by autofocus layers with different autofocus settings. We evaluated the proposed method with different settings on the BraTS 2018 dataset to investigate the impact on the segmentation performance and compared the results with related architectures. Results show that with our best models we achieved at least similar results as the models proposed in literature, with an average dice score of 83.92 for the binary case using 3D Autofocus Net. Due to compute and time constraints we were unable to search through the whole hyper-parameter space of the number of parallel convolutions and dilation rates within each layer. However it appears that an increase in the number of parallel convolutions in the autofocus layers improves the models and therefore further investigation into optimising this further is still required.

**Acknowledgement.** We would like to thank NVIDIA corporation, that donated a GPU to our group enabling this research. We would also like to thank the Engineering and Physical Sciences Research Council Impact Acceleration Account (EPSRC IAA) for supporting the initial stages of this work.

## References

1. Bakas, S., et al.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. *Sci. Data* **4**, 170117 (2017). <https://doi.org/10.1038/sdata.2017.117>
2. Blionas, A., Giakoumettis, D., Klonou, A., Neromyliotis, E., Karydakis, P., Themistocleous, M.S.: Paediatric gliomas: diagnosis, molecular biology and management. *Ann. Transl. Med.* **6**(12), 251 (2018). <https://doi.org/10.21037/atm.2018.05.11>
3. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-net: learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) *MICCAI 2016*. LNCS, vol. 9901, pp. 424–432. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46723-8\\_49](https://doi.org/10.1007/978-3-319-46723-8_49)
4. Davis, M.E.: Glioblastoma: overview of disease and treatment. *Clin. J. Oncol. Nurs.* (2016). <https://doi.org/10.1188/16.CJON.S1.2-8>
5. Havaei, M., et al.: Brain tumor segmentation with Deep Neural Networks. *Med. Image Anal.* **35**, 18–31 (2017). <https://doi.org/10.1016/j.media.2016.05.004>
6. Isensee, F., et al.: Abstract: nnU-Net: self-adapting framework for U-net-based medical image segmentation. *Bildverarbeitung für die Medizin 2019. I*, p. 22. Springer, Wiesbaden (2019). [https://doi.org/10.1007/978-3-658-25326-4\\_7](https://doi.org/10.1007/978-3-658-25326-4_7)
7. Kamnitsas, K., et al.: DeepMedic for brain tumor segmentation. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Winzeck, S., Handels, H. (eds.) *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2016*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10154, pp. 138–149. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-55524-9\\_14](https://doi.org/10.1007/978-3-319-55524-9_14)

8. Kamnitsas, K., et al.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Med. Image Anal.* **36**, 61–78 (2017). <https://doi.org/10.1016/j.media.2016.10.004>
9. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic gradient descent. In: *ICLR: International Conference on Learning Representations* (2015)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nat. Methods* **13**(1), 35 (2015). <https://doi.org/10.1038/nmeth.3707>
11. Li, W., Wang, G., Fidon, L., Ourselin, S., Cardoso, M.J., Vercauteren, T.: On the compactness, efficiency, and representation of 3D convolutional networks: brain parcellation as a pretext task. In: Niethammer, M., et al. (eds.) *IPMI 2017. LNCS*, vol. 10265, pp. 348–360. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-59050-9\\_28](https://doi.org/10.1007/978-3-319-59050-9_28)
12. Litjens, G., et al.: A survey on deep learning in medical image analysis. *Med. Image Anal.* (2017). <https://doi.org/10.1016/j.media.2017.07.005>
13. Liu, J., Li, M., Wang, J., Wu, F., Liu, T., Pan, Y.: A survey of MRI-based brain tumor segmentation methods (2014). <https://doi.org/10.1109/TST.2014.6961028>
14. Menze, B.H., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Med. Imaging* **34**(10), 1993–2024 (2015). <https://doi.org/10.1109/TMI.2014.2377694>
15. Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: fully convolutional neural networks for volumetric medical image segmentation. *J. Phys. E: Sci. Instrum.* **6**(9), 925–929 (2016). <https://doi.org/10.1088/0022-3735/6/9/035>
16. National Health Service. <https://www.nhs.uk/conditions/brain-tumours/>. Accessed 21 Mar 2020
17. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans. Med. Imaging* **35**(5), 1240–1251 (2016). <https://doi.org/10.1109/TMI.2016.2538465>
18. Qin, Y., et al.: Autofocus layer for semantic segmentation. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) *MICCAI 2018. LNCS*, vol. 11072, pp. 603–611. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-00931-1\\_69](https://doi.org/10.1007/978-3-030-00931-1_69)
19. NiftyNet. <http://www.niftynet.io/>. Accessed 6 June 2019
20. MICCAI BraTS’18 Challenge. <https://www.med.upenn.edu/sbia/brats2018.html>. Accessed 6 June 2019
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) *MICCAI 2015. LNCS*, vol. 9351, pp. 234–241. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
22. Shen, D., Wu, G., Suk, H.I.: Deep learning in medical image analysis. *Annu. Rev. Biomed. Eng.* **19**, 221–248 (2017). <https://doi.org/10.1146/annurev-bioeng-071516-044442>
23. Wang, G., Li, W., Ourselin, S., Vercauteren, T.: Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In: Crimi, A., Bakas, S., Kuijf, H., Menze, B., Reyes, M. (eds.) *BrainLes 2017. LNCS*, vol. 10670, pp. 178–190. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-75238-9\\_16](https://doi.org/10.1007/978-3-319-75238-9_16)
24. Wang, G., Li, W., Ourselin, S., Vercauteren, T.: Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation. In: Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., van Walsum, T. (eds.) *BrainLes 2018. LNCS*, vol. 11384, pp. 61–72. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-11726-9\\_6](https://doi.org/10.1007/978-3-030-11726-9_6)

25. Wirsching, H.G., Galanis, E.: Glioblastoma. In: Handbook of Clinical Neurology (2016). <https://doi.org/10.1016/B978-0-12-802997-8.00023-2>. Chapter 23
26. Yan, Y., Guo, Z.W., Zhang, H.B., Wang, N., Xu, Y.: Precision radiotherapy for brain tumors: a 10-year bibliometric analysis. *Neural Regeneration Res.* (2012). <https://doi.org/10.3969/j.issn.1673-5374>
27. Zikic, D., Ioannou, Y., Brown, M., Criminisi, A.: Segmentation of brain tumor tissues with convolutional neural networks. In: Proceedings MICCAI-BRATS (2014). <https://doi.org/10.1021/ol2032625>