



Contextual Additive Networks to Efficiently Boost 3D Image Segmentations

Zhenlin Xu^(✉), Zhengyang Shen, and Marc Niethammer

Department of Computer Science, UNC Chapel Hill, Chapel Hill, USA
zhenlinx@cs.unc.edu

Abstract. Semantic segmentation for 3D medical images is an important task for medical image analysis which would benefit from more efficient approaches. We propose a 3D segmentation framework of cascaded fully convolutional networks (FCNs) with contextual inputs and additive outputs. Compared to previous contextual cascaded networks the additive output forces each subsequent model to refine the output of previous models in the cascade. We use U-Nets of various complexity as elementary FCNs and demonstrate our method for cartilage segmentation on a large set of 3D magnetic resonance images (MRI) of the knee. We show that a cascade of simple U-Nets may for certain tasks be superior to a single deep and complex U-Net with almost two orders of magnitude more parameters. Our framework also allows greater flexibility in trading-off performance and efficiency during testing and training.

1 Introduction

Recently, deep convolution neural networks (CNNs) have shown excellent performance on various computer vision and medical image analysis tasks including semantic segmentation [1]. Early CNN approaches use sliding windows and approach segmentation as many independent classifications, which is inefficient. Fully-convolutional networks (FCN) [2] instead directly operate on full images. Consequentially, FCNs are more efficient and many FCN variants achieve state-of-the-art segmentation performance [3, 4]. When dealing with 3D image segmentations, the simplest approach is to treat a 3D volume as a sequence of 2D slices [5] and to segment them independently with a 2D CNN. However, this overlooks correlations across slices. To account for such correlations while avoiding 3D CNNs, triplanar schemes [6] have been proposed which apply 2D CNNs on image slices from three orthogonal planes of an image volume. Naturally, applying a 3D CNN to an image volume can take advantage of the full 3D information, but has high computational cost and memory requirements.

Most existing work on semantic segmentation focuses on improving performance by designing deeper and more complex networks. This, generally results in better performance, but comes at the cost of additional complexity, especially for the segmentation of 3D images. Hence, it would be beneficial to design more

efficient network architectures for 3D segmentation while *retaining* segmentation performance. Inspired by work that applies an auto-context approach [7] to CNN models [8] and additive learning schemes such as boosting [9], we propose a cascaded 3D semantic segmentation framework composed of a sequence of 3D FCNs with contextual inputs and additive outputs. As an alternative design strategy to a monolithic complex deep FCN, we show that such a sequence of simpler and shallower FCNs achieves performance on par with a more complex network, but using two orders of magnitude less parameters. This approach also allows to trade-off model accuracy with run-time and memory requirements.

Contributions: (1) We show that a cascaded model composed of several simple FCNs can perform as well as a single complex FCN with almost two orders of magnitude more parameters, resulting in better computational efficiency. (2) Our additive model shows better performance than an auto-context approach using contextual input (i.e., segmentations) only without the additive strategy. (3) We provide an analysis to give insight into why the additive output helps refine the segmentation model. (4) Lastly, we evaluate our model on a relatively large knee MRI dataset from the Osteoarthritis Initiative for cartilage segmentation.

2 Methods

In this section we (1) introduce the two components of our cascaded framework: contextual input and additive output; (2) provide an analysis illuminating the effect of additive outputs; and (3) describe the FCNs used to construct the cascaded models in our experiments. Figure 1 illustrates the proposed approach.

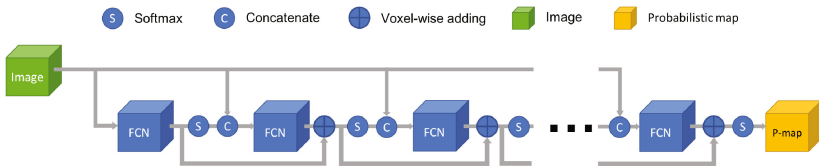


Fig. 1. Schematic diagram of proposed contextual additive model.

2.1 Contextual Additive Networks

Context information is useful for image segmentation [7, 10]. Inspired by the auto-context algorithm [7], cascaded models have been proposed that input the *concatenation* of an image and a segmentation (either the resulting labeling itself or the class label probabilities) to subsequent models. The segmentation is generated by a previous model with the image as its only input. Furthermore, residual skip connections [11] are widely used for CNNs. These help the training of deep networks and boost performance. Our contextual additive network is inspired by both approaches. However, instead of using the residual connections

across layers inside a neural network, we use them to connect the output of each sub-model to generate the class probability. We use a sequence of such models each also having access to the original input image (see Fig. 1).

Formally, our cascaded model Φ is based on a sequence of FCNs $\{\phi^0, \phi^1, \dots, \phi^M\}$, whose parameters are $\Theta = \{\theta^0, \theta^1, \dots, \theta^M\}$ respectively. The first FCN, ϕ^0 , with parameters θ^0 takes an image x as input and predicts the probability map of all class labels, P^0 , by applying softmax to the output of the FCN: $P^0(x; \theta^0) = \sigma(\phi^0(x; \theta^0))$, where σ is the softmax function. For an output $z \in \mathbb{R}^C$ of C classes, the probability of class j is

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{l=0}^{C-1} e^{z_l}}, \quad c \in \{0, \dots, C-1\}. \quad (1)$$

Subsequent FCNs use the image and the probability map (i.e., the *contextual input*) of the previous FCN as input. However, instead of directly predicting the input to a softmax function to obtain label probabilities these subsequent FCNs (unlike previous work [8]) predict a residual between the previous prediction, added to the output of the previous stage (i.e., the *additive output*) before the softmax. The output of the contextual additive model after the M -th FCN is

$$P^M(x; \Theta) = \sigma(\phi^0(x; \theta^0)) + \sum_{m=1}^M \phi^m(x, P^{m-1}; \theta^m). \quad (2)$$

Such a cascaded model can be trained by training each additive FCN via:

$$\hat{\theta}^m = \arg \min_{\theta^m} \mathcal{L}(Y, P^m(X; \{\hat{\theta}^0, \dots, \hat{\theta}^{m-1}, \theta^m\})), \quad (3)$$

where Y denotes the set of label images, X the set of images in the training dataset, and \mathcal{L} is the chosen loss function. Alternatively it can be trained end-to-end by minimizing the sum of the losses for all stages of the model:

$$\hat{\Theta} = \{\hat{\theta}^0, \hat{\theta}^1, \dots, \hat{\theta}^M\} = \arg \min_{\Theta} \sum_{m=0}^M \mathcal{L}(Y, P^m(X; \{\theta^0, \dots, \theta^m\})). \quad (4)$$

Both training strategies work well in our experiments. When applying the trained model one obtains the class label by selecting the most probable label:

$$\hat{y}(x; \hat{\Theta}) = \arg \max_j P_j^M(x; \hat{\Theta}), \quad (5)$$

where \hat{y} denotes the label output for input image x and model parameters $\hat{\Theta}$.

2.2 Why an Additive Network is Beneficial

To give insight into the effect of adding model outputs *before* the softmax in the cascade we approximate the loss function to first order. We use the cross-entropy loss for multi-class segmentation which for a single model output, ϕ^0 , is

$$\mathcal{L}_{CE}^0 = - \sum_{j=0}^{C-1} y_j \ln(\sigma(\phi_j^0)), \quad (6)$$

where j is the class index and C is the total number of classes. Considering a cascaded model of two FCNs, we assume we trained the first FCN ϕ^0 by optimizing \mathcal{L}_{CE}^0 . With the additive output of the second model, the loss becomes

$$\mathcal{L}_{CE}^1 = - \sum_{j=0}^{C-1} y_j \ln(\sigma(\phi^0 + \phi^1)_j). \quad (7)$$

We can think of ϕ^1 as a perturbation to ϕ^0 . Approximating the loss function (7) around ϕ^0 via a Taylor series expansion results in

$$\begin{aligned} \mathcal{L}_{CE}^1 &\approx - \sum_{j=0}^{C-1} y_j \ln(\sigma(\phi_j^0)) - \sum_{j=0}^{C-1} y_j \sum_{l=0}^{C-1} \frac{\partial \ln \sigma(\phi^0)_j}{\partial \phi_l^0} \phi_l^1 \\ &= \mathcal{L}_{CE}^0 + \sum_{j=0}^{C-1} y_j \sum_{l=0}^{C-1} \Delta \mathcal{L}_{CE_j}^1(\phi_l^1 | \phi^0), \end{aligned} \quad (8)$$

where \mathcal{L}_{CE}^0 only depends on ϕ^0 and can therefore be ignored for sequential training of ϕ^1 ; $\Delta \mathcal{L}_{CE_j}^1(\phi_l^1 | \phi^0)$ captures how the loss depends on the output of the second model for class l , ϕ_l^1 , for voxels annotated as class j :

$$\Delta \mathcal{L}_{CE_j}^1(\phi_l^1 | \phi^0) = \begin{cases} -(1 - \sigma(\phi_j^0)) \phi_j^1 = -(1 - P_j^0) \phi_j^1, & l = j \\ \sigma(\phi_l^0) \phi_l^1 = P_l^0 \phi_l^1, & l \neq j \end{cases}. \quad (9)$$

Intuitively, when the first model performs well P_j^0 is high and $P_{l,l \neq j}^0$ is low; increasing ϕ_j^1 and decreasing $\phi_{l,l \neq j}^1$ is of low benefit to reduce the loss. When the first model performs badly P_j^0 is low and $P_{l,l \neq j}^0$ is high; increasing ϕ_j^1 and decreasing $\phi_{l,l \neq j}^1$ is of high benefit. I.e., improving the prediction where the first model perform badly is more beneficial than improving already good predictions. In effect, the loss of the additive model naturally weighs each voxel so that it focuses on problematic regions.

2.3 3D Fully Convolution Networks

Many FCN variants exist [3, 12]. The U-Net [13] and the 3D U-Net [14] have been popular to segment medical images. U-Nets add skip connections between the encoder/decoder paths to retain high resolution features. We use the 3D U-Net as our elementary FCN because of its good performance. The original 3D U-Net is a dense architecture with four resolution steps in the encoder/decoder paths, and 512 feature channels at the bottleneck, resulting in a total of ~ 19 million parameters. We also build three simpler U-Nets with fewer feature channels and fewer resolution levels (Fig. 2). The smallest one has only 45,808 parameters.

3 Experiments

For each U-Net, we train a cascaded model of length M , where M is larger for smaller U-Nets as the performance of a model with more complex U-Nets saturates with smaller M . We explore results for end-to-end and sequential training.

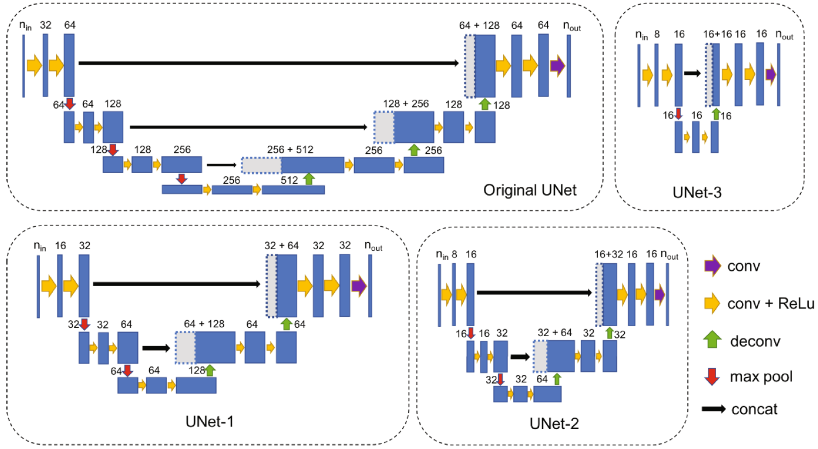


Fig. 2. U-Nets of the cascaded models (# of parameters in parentheses): original U-Net (~19M), U-Net-1 (~1.1M), U-Net-2 (~287K), U-Net-3 (~46K)

We also use only contextual input and only additive output for our cascaded U-Net-3 × 6 model to investigate the impact of our two key techniques. We study memory use and runtime to explore our model’s segmentation efficiency.

3.1 Dataset and Preprocessing

We use knee MRIs from the Osteoarthritis Initiative consisting of 176 MR images from 88 patients (2 longitudinal scans per patient). We split the dataset into a training set of 60 patients (120 images), a validation set of 8 patients (16 images) and a test set of 20 patients (40 images). All images are of size 384 × 384 × 160 and resolution 0.36 × 0.36 × 0.7 mm³ per voxel. We normalize the intensities of each image such that the 0.1 percentile and the 99.9 percentile are mapped to {0, 1} and clamp values that are smaller to 0 and larger to 1 to avoid outliers. We did not apply bias-field correction, because our exploratory experiments indicated that bias-field correction did not substantially impact segmentation results. For each volume, femoral and tibial cartilage are annotated on sagittal slices. We transform the corresponding 2D polygon annotations to 3D label maps.

Table 1. Models’ parameter size and memory consumption in sequential training

Model		Original U-Net	U-Net-1 × 2	U-Net-2 × 3	U-Net-3 × 6
params #		19,065,888	2,294,486	862,185	275,538
Memory (MB)	Train	11116	5836	3190	2434
	Test	10312	7614	4044	2820

3.2 Implementation Details

Due to the high memory demands of 3D convolutions, the full image volume and its network outputs may not fit on a single GPU. Hence, we use overlapping tiles as in the U-Net [13]. We choose image patches of size $128 \times 128 \times 32$ considering the nonuniform voxel resolution and that annotations were drawn sagittally.

During training, we randomly crop 3D patch pairs from image-label pairs. To avoid class imbalances due to the high proportion of background voxels we use three types of patches: any possible patch, patches with more than $r_1\%$ of femoral cartilage voxels, and patches with more than $r_2\%$ tibial cartilage voxels. Patches are randomly sampled at a ratio of $1 : 1 : 2$ ($r_1 = 1$, $r_2 = 2$). We use the Adam [15] optimizer with first moment $\beta_1 = 0.9$, second moment $\beta_2 = 0.999$, and $\epsilon = 1e-10$. The learning rate is initialized as $5e-4$ and decays at half of the total epochs and at the beginning of the last 50 epochs by 0.2. We train the original U-Net and each sub-network in the sequentially trained cascaded models with 600 epochs. When training a cascaded model of M U-Nets end-to-end, $100 * (M - 1)$ extra epochs were applied to assure convergence. Regarding training time, the cascaded models take less time than the original U-Net (13 h) except U-Net- 3×6 (17 h for end-to-end training and 20 h for sequential training). During training, we recorded a model’s Dice score on the validation dataset and evaluate the model with the best validation score on the separate testing dataset.

Table 2. Segmentation evaluation of contextual additive models using different U-Nets. E.g. U-Net- 1×2 is a cascaded model of two U-Net-1. Results are for sequential training (end-to-end results in parentheses). Our models can achieve performance on par with the original U-Net with much fewer parameters and lower memory requirements.

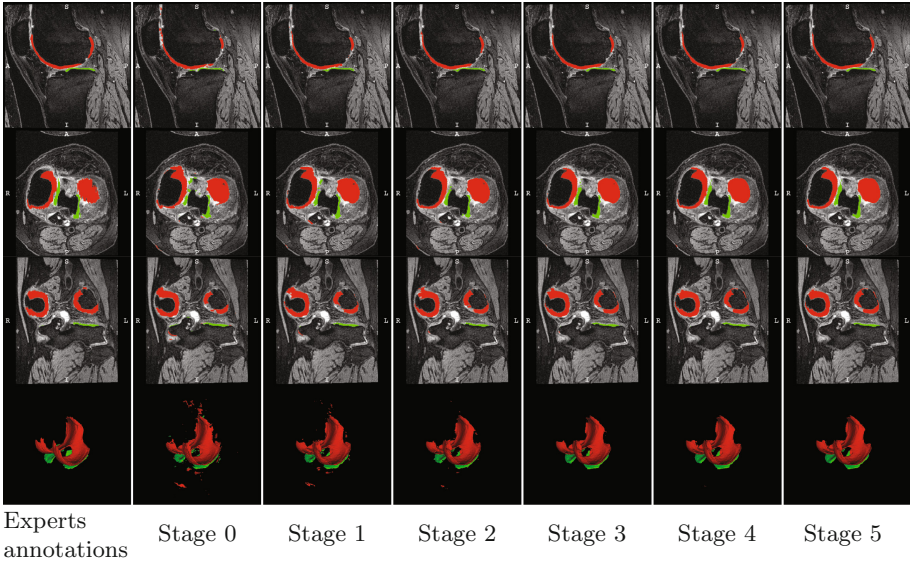
Model	Stage	DSC (%)	mIOU (%)
Original U-Net	-	89.08 ± 2.41	86.89 ± 2.56
U-Net- 1×2	0	88.88 ± 2.61 (88.78 ± 2.60)	86.69 ± 2.76 (86.58 ± 2.74)
	1	89.17 ± 2.55 (89.31 ± 2.39)	87.00 ± 2.71 (87.15 ± 2.55)
UNet- 2×3	0	88.13 ± 2.55 (88.31 ± 2.60)	85.88 ± 2.67 (86.07 ± 2.72)
	1	88.72 ± 2.47 (88.79 ± 2.34)	86.50 ± 2.61 (86.58 ± 2.47)
	2	88.74 ± 2.51 (89.14 ± 2.30)	86.53 ± 2.66 (86.96 ± 2.45)
UNet- 3×6	0	85.00 ± 3.13 (83.44 ± 3.00)	82.64 ± 3.10 (81.08 ± 2.90)
	1	87.68 ± 2.66 (86.83 ± 2.68)	85.40 ± 2.77 (84.51 ± 2.73)
	2	88.23 ± 2.50 (88.08 ± 2.57)	85.98 ± 2.62 (85.83 ± 2.68)
	3	88.57 ± 2.45 (88.70 ± 2.45)	86.34 ± 2.59 (86.48 ± 2.59)
	4	88.63 ± 2.42 (89.01 ± 2.35)	86.40 ± 2.57 (86.81 ± 2.50)
	5	88.67 ± 2.42 (89.10 ± 2.35)	86.45 ± 2.56 (86.92 ± 2.50)
UNet- 3×6 (contextual input only)	5	88.23 ± 2.59	85.98 ± 2.71
UNet- 3×6 (additive output only)	5	87.22 ± 2.78	84.92 ± 2.87

4 Results and Discussion

We quantitatively evaluate the segmentation results of each model and the output at intermediate stages. Table 2 shows average Dice scores (DSC) and the mean Intersection of Union (mIOU) of femoral and tibial cartilage and their standard deviations. We also report the performance of U-Net- 3×6 models using contextual input or additive output only. The number of model parameters and memory consumption in sequential training (batch size 4) and testing (batch size 8) are given in Table 1. Table 3 shows segmentation results at different stages of the U-Net- 3×6 cascade.

We observe that our contextual additive networks are more efficient as they use significantly fewer parameters while achieving similar or better performance than using a single more complex U-Net. The original U-Net has for example almost two orders of magnitude more parameters than the U-Net- 3×6 while resulting in very similar accuracy. We also observe that both the contextual inputs *and* the additive output helps boost the performance in cascaded U-Nets.

Table 3. Segmentation results of end-to-end trained U-Net- 3×6 . Rows are Sagittal, Axial, Coronal views and 3D rendering. Red and green labels represent femoral and tibial cartilage respectively.



5 Conclusion

We developed a framework of cascaded FCNs with contextual inputs and additive output to boost the performance of 3D semantic segmentation. Our theoretical

analysis shows that the additive output focuses the additive model on regions where previous output results were relatively poor. Experiments on a large 3D MRI knee dataset demonstrated that our framework can refine the results of a single U-Net. Importantly, we showed that a cascaded model of simple U-Nets can match the performance of a complex U-Net, while providing better efficiency in terms of using fewer parameters and requiring less memory. Our approach may provide an alternative to improve FCNs for segmentation. Future work will investigate different FCNs as elements of the cascade, e.g. networks with inputs of multiple resolutions, and evaluate performance on different datasets.

References

1. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Garcia-Rodriguez, J.: A review on deep learning techniques applied to semantic segmentation. arXiv preprint [arXiv:1704.06857](https://arxiv.org/abs/1704.06857) (2017)
2. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR, pp. 3431–3440 (2015)
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS. arXiv preprint [arXiv:1606.00915](https://arxiv.org/abs/1606.00915) (2016)
4. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 3DV, pp. 565–571. IEEE (2016)
5. Chen, H., Qi, X., Yu, L., Dou, Q., Qin, J., Heng, P.A.: DCAN: deep contour-aware networks for object instance segmentation from histology images. *Med. Image Anal.* **36**, 135–146 (2017)
6. Prasoon, A., Petersen, K., Igel, C., Lauze, F., Dam, E., Nielsen, M.: Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8150, pp. 246–253. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40763-5_31
7. Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *TPAMI* **32**(10), 1744–1757 (2010)
8. Salehi, S.S.M., Erdogmus, D., Gholipour, A.: Auto-context convolutional neural network (auto-net) for brain extraction in magnetic resonance imaging. *IEEE Trans. Med. Imaging* **36**(11), 2319–2330 (2017)
9. Friedman, J., Hastie, T., Tibshirani, R., et al.: Additive logistic regression: a statistical view of boosting. *Ann. Stat.* **28**(2), 337–407 (2000)
10. Loog, M., van Ginneken, B.: Supervised segmentation by iterated contextual pixel classification. In: 16th International Conference on Pattern Recognition. Proceedings, vol. 2, pp. 925–928. IEEE (2002)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
12. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *TPAMI* **39**(12), 2481–2495 (2017)
13. 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

14. 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
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)