# Full Volumetric Brain Tissue Segmentation in Non-Contrast CT using Memory Efficient Convolutional LSTMs

**Sil C. van de Leemput, Ajay Patel, Rashindra Manniesing**
Department of Radiology and Nuclear Medicine
Radboud University Medical Center
6525 GA Nijmegen, The Netherlands
sil.vandeleemput@radboudumc.nl

## Abstract

There is a demand for deep learning approaches able to process high resolution 3D volumes in an accurate and fast way. However, training of these models is often limited by the available GPU memory, which often results in reduced model depth, receptive field, and input size, limiting the expressiveness of the model. In this work we present a memory efficient modified convolutional-LSTM, which integrates a context-rich 2D U-Net as an input in a slice based manner and subsequently integrates the acquired slices using LSTM to create the full 3D context. Memory savings achieved by checkpointing on one or more steps within the LSTM allow for direct training on a single full non-contrast CT volume of: 512 x 512 x 320 on a NVIDIA Titan X with 12 GB of VRAM. We demonstrate the effectiveness of our method by training and segmenting the cranial cavity including soft-brain tissue and CSF in the non-contrast CT end-to-end on the full image data, without any stitching, while preserving a large receptive field and high expressiveness.

## 1 Introduction

Developing deep learning methods for full 3D segmentation of high resolution images is foremost challenging due to the GPU memory requirements during training and inference. Designing models for such applications often requires trading-off model expressiveness, like number of layers, input size and others to free up memory. The tiling and stitching technique as described in [1] is a popular way of processing an image in smaller chunks to save memory, but limits the size of the receptive field and hence the context the network can capture.

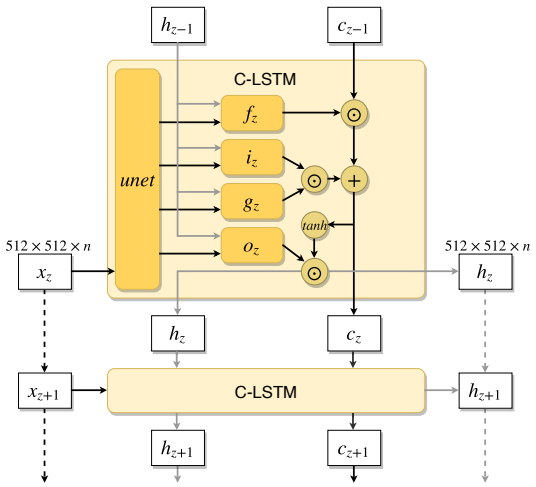
We propose a novel memory efficient convolutional LSTM architecture utilizing checkpointing techniques [2] with an integrated 2D U-Net [1] to enable high resolution whole volume level training and prediction, with reasonable training and inference times.

Similar to the work of Chen et al. [3] we employ convolutional LSTM and 2D U-Net in conjunction to segment 3D volumetric data. However, in our case we have explicitly integrated the 2D U-Net inside the C-LSTM to further push down memory requirements per slice. Furthermore, we are able to fit much larger volumes due to memory savings by utilizing checkpointing.

## 2 Method

The model used in this work consists of a modified convolutional LSTM (C-LSTM) [4] which is essentially a LSTM [5] with the dot products within the gates replaced with convolution operations. Additionally, our model replaces the single layer input convolution with a 2D U-Net [1]

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.




$$
\begin{aligned}
u_z &= unet(x_z, W_{unet}) \\
i_z &= \sigma(u_{z,i} + h_{z-1} * W_i + b_i) \\
f_z &= \sigma(u_{z,f} + h_{z-1} * W_f + b_f) \\
o_z &= \sigma(u_{z,o} + h_{z-1} * W_o + b_o) \\
g_z &= tanh(u_{z,c} + h_{z-1} * W_c + b_c) \\
c_z &= f_z \odot c_{z-1} + i_z \odot g_z \\
h_z &= o_z \odot tanh(c_z)
\end{aligned}
\tag{1}
$$

Figure 1: The modified C-LSTM model. On the right the formulas describing the model, where $\sigma$ is the sigmoid function, $tanh$ is the hyperbolic tangent function, $*$ is a convolution operation, $\odot$ is the element-wise sum or Hadamard product, and $unet(\cdot, W_{unet})$ is a 2D U-Net forward pass given network weights $W_{unet}$. Furthermore, $x_z$ and $h_z$ represent respectively the input slice and the internal hidden state for slice number $z$. The trainable parameters in the model are: the U-Net weights $W_{unet}$, the weights for the hidden-to-hidden convolutions $W_i, W_f, W_o, W_c$ and the biases $b_i, b_f, b_o, b_c$. On the left a visual representation of the C-LSTM model, illustrating how the model can be sequentially applied to generate the output $h_0, h_1, \ldots, h_n$ from inputs $x_0, x_1, \ldots, x_n$ respectively.

$unet(\cdot, W_{unet})$ for processing each axial slice $x_z$ from the input $x$ given U-Net weights $W_{unet}$. The full model is described by the equations in Figure 1. The model was implemented in PyTorch.

### 2.1 2D U-Net

The 2D U-Net used in this work has 4 max-pools and upscale operations and is mostly similar to the original implementation from [1], using an upward and downward path, long skip connections and the same number of filters at each layer. However, all convolutions in the network use 1 pixel border of zero padding such that the size of the input to the network matches the size of the output. Furthermore, batch normalization [6] was added after each convolution, but before the rectified linear units. Finally, nearest neighbor upsampling was used instead of deconvolution.

### 2.2 Memory saving

The equations from Figure 1 are typically rewritten as a step-function $step(x_z, h_{z-1}, c_{z-1}) = (h_z, c_z)$ which at each slice number $z$ takes input axial slice $x_t$ and previous hidden and cell states $h_{z-1}$, $c_{z-1}$ and produces the next hidden state $h_z$ and cell state $c_z$. By checkpointing [2] the step function, the internal activation maps are freed after computing the hidden and cell states and only the inputs $(x_z, h_{z-1}, c_{z-1})$ and outputs $(h_{z-1}, c_{z-1})$ activations are retained in memory. On the backward pass, having the input and outputs is sufficient to recompute the intermediate activation maps required for training. Since in this manner only the activation maps of one step function are fully in memory at a time, a significant reduction of the memory overhead is achieved during training at the expense of some additional computation time to recompute the activation maps.

## 3 Experiment and data

We validate our approach on a brain tissue segmentation task [7] dataset in non-contrast CT (NCCT) of the head, where each scan consists of a volume of $512 \times 512 \times$ 302-400 voxels in respectively coronal, sagittal, and axial direction. Note that in order to segment the brain, sufficient context is necessary to delineate intracranial brain tissue from extracranial brain tissue, thus making it a good candidate task to test our model.

Figure 2: Qualitative segmentation results of one of the test cases in the form of three cross sections. From left to right: a sagittal, coronal, and axial slice. A green overlay indicates in agreement with the reference standard (true positives). Red overlay indicates undersegmentation (false negatives) and purple indicates oversegmentation (false positives).

We took a total of 157 NCCT cases and splitted them into a training set of 77 cases, a validation set of 20 cases and a test set of 60 cases. For all cases the full brain mask was computed using the method of Patel et al. [7] as the reference standard. All NCCT data were normalized using z-score normalization with statistics computed beforehand over the training-set. The training data were augmented on the fly during training using random rotations $[-10, 10]^\circ$ and random translations $[-20, 20]$ voxels for each dimension.

### 3.1 Training

Essentially our network can be trained end-to-end, but in order to simplify and speed up training we employed a two-stage training scheme. In both training phases the binary cross entropy between prediction and the reference standard was picked as the loss function to minimize. First, the 2D U-Net was trained separately in a slice-based manner on the training-set for 1000 randomly picked slices with one slice per batch, using a stochastic gradient descent solver with learning rate of $0.01$ and a momentum factor of $0.99$. The resulting best performing training iteration on the validation set was taken to pre-initialize the U-Net within the modified C-LSTM model. Next, the full C-LSTM model was trained using the pre-initialization with an RMSprop optimizer for $1500$ iterations on a single full 3D NCCT case per iteration, with a learning rate of 1e-5 and a momentum of $0$. The pre-initialized weights were fixed for the first $1000$ iterations. All models were trained on a single NVIDIA Titan X with 12 GB of VRAM.

During training, every $25$ iterations the Dice score [8] was computed between the predictions and the reference standard on the validation set. The model at the iteration with the highest average Dice score of the C-LSTM model was selected to compute the scores on the test set.

## 4 Results

The average Dice results for the brain segmentation task on the test set of 60 cases were: $0.987 \pm 0.007$. A qualitative example of the segmentation results is shown in Figure 2. Training of a single volume of 320 axial slices used approximately 8 GB of VRAM and could be performed in under 105 seconds. Prediction of an entire NCCT took less than 45 seconds on the GPU.

## 5 Conclusion

We have presented a modified C-LSTM network with an integrated 2D U-Net and checkpointing which has low memory requirements, enables full 3D volume processing, and maintains reasonable training times and model expressiveness. We have demonstrated the potential of the model by training it end-to-end on high resolution 3D NCCT images to finally acquire high quality full resolution brain tissue segmentations without the need for tiling and stitching. In future work we would like to apply our model to different problems and compare it with the current state of the art.

## References


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

[2] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," *arXiv preprint arXiv:1604.06174*, 2016.

[3] J. Chen, L. Yang, Y. Zhang, M. Alber, and D. Z. Chen, "Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation," in *Advances in Neural Information Processing Systems*, 2016, pp. 3036–3044.

[4] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

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

[7] A. Patel, B. van Ginneken, F. J. Meijer, E. J. van Dijk, M. Prokop, and R. Manniesing, "Robust cranial cavity segmentation in CT and CT perfusion images of trauma and suspected stroke patients," *Medical image analysis*, vol. 36, pp. 216–228, 2017.

[8] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, pp. 297–302, 1945.