



A Joint 3D UNet-Graph Neural Network-Based Method for Airway Segmentation from Chest CTs

Antonio Garcia-Uceda Juarez^{1,2(✉)}, Raghavendra Selvan³, Zaigham Saghir⁴,
and Marleen de Bruijne^{1,3}

¹ Biomedical Imaging Group Rotterdam,
Department of Radiology and Nuclear Medicine, Erasmus MC,
3015 CE Rotterdam, The Netherlands
a.garciauced@erasmusmc.nl

² Department of Pediatric Pulmonology, Erasmus MC-Sophia Children Hospital,
3015 CE Rotterdam, The Netherlands

³ Department of Computer Science, University of Copenhagen,
2100 Copenhagen, Denmark

⁴ Department of Medicine, Section of Pulmonary Medicine, Herlev-Gentofte Hospital,
Copenhagen University Hospital, Kildegårdsvej 28, 2900 Hellerup, Denmark

Abstract. We present an end-to-end deep learning segmentation method by combining a 3D UNet architecture with a graph neural network (GNN) model. In this approach, the convolutional layers at the deepest level of the UNet are replaced by a GNN-based module with a series of graph convolutions. The dense feature maps at this level are transformed into a graph input to the GNN module. The incorporation of graph convolutions in the UNet provides nodes in the graph with information that is based on node connectivity, in addition to the local features learnt through the downsampled paths. This information can help improve segmentation decisions. By stacking several graph convolution layers, the nodes can access higher order neighbourhood information without substantial increase in computational expense. We propose two types of node connectivity in the graph adjacency: (i) one predefined and based on a regular node neighbourhood, and (ii) one dynamically computed during training and using the nearest neighbour nodes in the feature space. We have applied this method to the task of segmenting the airway tree from chest CT scans. Experiments have been performed on 32 CTs from the Danish Lung Cancer Screening Trial dataset. We evaluate the performance of the UNet-GNN models with two types of graph adjacency and compare it with the baseline UNet.

Keywords: Convolutional neural networks · Graph neural network · Graph convolution · Airway segmentation

1 Introduction

Since recently, fully convolutional neural networks (CNNs) are the state-of-the-art for many segmentation tasks [1], and in particular the UNet architecture [2] for biomedical image segmentation. The UNet consists of an encoding path, in which high-order features are extracted at several downsampled resolutions, followed by a decoding path, in which these features are leveraged to the full resolution to perform voxel-wise segmentation decisions. An extension of CNNs to graph structured data are Graph neural networks (GNNs) [3, 4], which have seen early application for segmenting structures that resemble graphs [8, 9]. Initial work of combining CNNs and GNNs was by proposed Shin et al. [9] for 2D vessel segmentation. This was a sequential pipeline in which the CNN was trained for feature extraction prior to applying the GNNs to learn global connectivity.

The segmentation of tree-like structures such as the airways in chest CTs is a complex task, with branches of varying sizes and different orientations while taking into account bifurcations. A comparison of airway extraction methods (prior to CNNs) was performed in the EXACT challenge [5]. The results showed that all methods missed a significant amount of the small, peripheral branches. The UNet has since been applied for airway segmentation in [6, 7]. The 3D UNet-based method in [7] is fully automatic and can segment the airways in a full lung in a single pass through the network. However, this method had problems to capture various small terminal branches. Also, the GNNs have been applied to airway extraction as a graph refinement approach in [8]. However, this method was not end-to-end optimised and relied on handcrafted features as input to the graph.

In this paper, we present an end-to-end segmentation approach by combining a 3D UNet with GNNs and evaluate this on extracting the airway tree from chest CTs. This method replaces the two convolutional layers in the deepest level of the baseline UNet by a GNN-based module, comprising graph convolutions. This end-to-end approach simultaneously learns local image features and global information based on graph neighbourhood connectivity.

2 Methods

The proposed joint UNet-GNN architecture is described in the following subsections. This approach integrates a GNN module at the deepest level of a baseline 3D UNet, and is schematically shown in Fig. 1 (left). The GNN module uses a graph structure obtained from the dense feature maps resulting from the contracting path of the Unet. Each graph node can be viewed as a “supervoxel” from the downsampled regions with the corresponding vector of features. The connectivity of nodes in such a graph is described by the adjacency matrix and determines the neighbourhood of each node when performing graph convolutions, as shown in Fig. 1 (right). The GNN module learns combinations of the input feature maps based on the graph topology, and outputs another graph with same “supervoxel” nodes as the input graph and the corresponding vector

of learnt features for each node. This output are feature maps that are fed to the upsampling path of the UNet.

We have tested two types of node connectivity that are: (i) Predefined and based on a regular neighbourhood, with each node connected to its 26-direct neighbours, and (ii) Dynamically computed during training based on choosing nearest neighbours in the node feature space. Stacking several GNN layers allows nodes to access longer range information beyond the initial neighbourhood, which can improve the segmentation decisions as more complex features that include relevant information from nodes far away in the volume can be used. This is in contrast with CNNs which rely on local feature extraction, and access long range information via successive convolutions and pooling, with a detriment of strong reduction in resolution. This long range access by the GNN module is useful when segmenting tree structures like airways, as information from branches that share some features (shape, orientation, ...) but are further away in the volume can be used when detecting a given branch. The computation of the dynamic graph adjacency is further explained in Sect. 2.2.

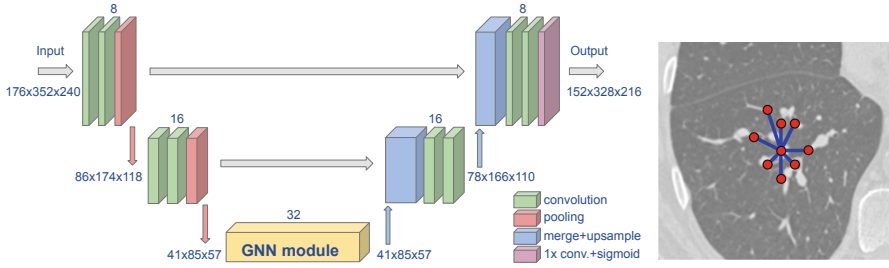


Fig. 1. Left: schematics of a UNet-GNN network of 3 levels. Right: illustration of irregular node connectivity for a given voxel in the initial graph.

2.1 Graph Neural Network (GNN) Module

The main component of the GNN module is a series of graph convolutional layers [3, 4]. This operation can be seen as a generalisation of the cartesian convolution to a graph setting. One of the formulations of graph convolution operation with separate processing of self connections proposed in Kipf et al. [4] is given by the equation,

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{H}^{(l)}\mathbf{W}_0^{(l)} + \mathbf{D}^{-1}\mathbf{A}\mathbf{H}^{(l)}\mathbf{W}_1^{(l)}\right) \quad (1)$$

where $\sigma(\cdot)$ is the rectified linear unit activation function, $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times E}$ is the node feature matrix comprising N nodes and E features input to the l^{th} GNN layer. The learnable GNN filter weights are $\mathbf{W}_0^{(l)}, \mathbf{W}_1^{(l)} \in \mathbb{R}^{E \times E}$. $\mathbf{A} \in \{0, 1\}^{N \times N}$ is the binary adjacency matrix, \mathbf{D} is the degree matrix derived from \mathbf{A} with

diagonal entries $D_{ii} = \sum_{j=1}^N A_{ij}, \forall i = 1 \dots N$. The adjacency matrix is largely sparse, with non-zero entries per node corresponding to the size of each node neighbourhood (26 for the regular neighbourhood case above). By processing the adjacency matrix as a sparse tensor, operations in Eq. (1) are done efficiently.

The GNN module in the proposed method has $L = 4$ layers performing the operations in Eq. (1) successively. By stacking several graph convolutions together, each node in the output graph updates its features with information from higher order neighbourhood, which can improve the segmentation decisions. The initial graph features $\mathbf{H}^{(0)}$ are obtained as,

$$\mathbf{H}^{(0)} = f(\mathbf{Z}) \quad (2)$$

where $\mathbf{Z} \in \mathbb{R}^{N \times F}$ is the F -dimensional node feature matrix derived from the UNet and $f(\cdot)$ is a two layered multi-layer perceptron with F input units and E output units and rectified linear units, followed by a normalization layer. The transformation in Eq. (2) allows more complex representations of the input node features useful for the GNN module.

2.2 Irregular Neighbourhood

A GNN module with regular adjacency has limited node connectivity comprising only the direct neighbours. This constraint is imposed primarily to keep the adjacency matrix sparse to reduce the large memory footprint. In order to allow nodes in the graph to access information well beyond their directly connected neighbours, we propose an extension to the GNN module in Sect. 2.1 by using a graph adjacency with node connectivity as the k – nearest neighbours in the feature space. Node neighbourhood is decided from the pairwise euclidean distance between nodes from \mathbf{Z} in the F – dimensional feature space. In this work, we set the number of neighbours to $k = 26$. The graph adjacency is dynamically computed during training for every input image in every epoch. This approach, we argue, enables the method to access irregular but meaningful neighbourhoods and utilises the inherent capabilities of using GNN-based learning over irregular neighbourhoods in an improved fashion.

3 Experiments

3.1 Dataset

We used 32 low-dose CT chest scans from the Danish Lung Cancer Screening Trial [13]. All scans have voxel-resolution of roughly $0.78 \times 0.78 \times 1 \text{ mm}^3$. The reference segmentations are airway lumen obtained from the method [10] applied on the union of two previous methods [11, 12], and corrected by an expert observer.

3.2 Network Implementation

The baseline UNet upon which the UNet-GNN approach is built derives from the model in [7] with 3 levels of resolution. Each level in the downsampling/upsampling path is composed of two $3 \times 3 \times 3$ convolutional layers with rectified linear (ReLU) activation, and followed by a $2 \times 2 \times 2$ pooling/upsample layer, respectively. No padding is used in the convolutions in order to reduce the model memory footprint. There are 8 feature maps in the first level, and this is doubled/halved after every pooling/upsampling, respectively. The GNN module in the deepest level has twice as many output features per voxel as input units, i.e. $E = 2F$ in Eqs. (1) and (2). The final layer consists of a $1 \times 1 \times 1$ convolutional layer followed by a sigmoid activation function.

A series of operations to extract the input images from the volume CTs as in [7] are applied. These consist of: (i) crop the CTs to a bounding-box around pre-segmented lung fields, (ii) extract smaller input images in a sliding-window fashion in axial dimension, and (iii) apply image rigid transformation for data augmentation. The model is designed for the largest input images that can fit in GPU memory. This corresponds to a size $176 \times 352 \times 240$ in a batch containing only one image, for a GPU NVIDIA GeForce GTX 1080 Ti with 11 GB memory used in these experiments. The models were implemented in Pytorch framework.

3.3 Training the Models

The loss function used for training the network is the dice loss,

$$\mathcal{L} = \frac{2 \sum_{x \in N_L} p(x)g(x)}{\sum_{x \in N_L} p(x) + \sum_{x \in N_L} g(x) + \epsilon} \quad (3)$$

where $p(x)$, $g(x)$ are the predicted voxel-wise airway probability maps and airway ground truth, respectively. The ground truth is masked to the region of interest (ROI): the lungs, indicated by the sub-index L , so that only voxels within this region contribute to the loss. This mask removes the trachea and main bronchi from the ground truth, so that training is focused on the smaller branches. The lung segmentation needed for this masking operation is easily computed by a region-growing method in [11]. ϵ is a tolerance needed when there are no ground truth voxels in the image patch.

To train the networks we use 16 CTs, 4 CTs for model validation/hyperparameter tuning, and the remaining 12 CTs for testing. The Adam optimizer is used with a learning rate that is chosen for each model as large as possible while ensuring convergence of the losses. This was 10^{-4} for UNet-GNN models and 5×10^{-5} for UNet models. All models are trained until convergence, and we retrieve the model with overall minimum validation loss for testing. As convergence criterion, we monitor the moving average of the validation loss over 50 epochs, and training is stopped when its value (i) increases by more than 5%, or (ii) does not decrease more than 0.1%, over 20 epochs (patience). Training time was approximately 1–2 days, depending on the models, while test time inference takes a few seconds per scan.

3.4 Experiments Set-Up

We evaluate four different models: two regular UNets with 3 resolution levels (UNetLev3) and 5 levels (UNetLev5), and two UNet-GNN models each with 3 levels. The UNet-GNN models differ in the type of graph neighbourhood: (i) regular graph adjacency with 26 direct neighbours (UGnnReg), and (ii) dynamic computation of the adjacency matrix (UGnnDyn), with 26 connections per node as described in Sect. 2.2. With our experiments we evaluate: (i) the benefit of the GNN module at deepest level of the UNet when compared to the two convolutional layers of a regular UNet, and (ii) the difference in performance of UNet-GNN models when compared to a more complex model like the UNetLev5.

The models are compared based on (i) Dice overlap coefficient, (ii) airway completeness, (iii) volume leakage, and two centreline distance error measures, (iv) false negative d_{FN} and (v) false positive d_{FP} distances, as in [8]. Completeness is the ratio of ground truth centreline length inside the predictions with respect to ground truth centreline full length, and volume leakage is the ratio of number of false positive voxels with respect to the number of voxels in the ground truth. We refer the reader to [8] for the definition of d_{FN} and d_{FP} . Trachea and main bronchi are removed in these measures from both predictions and ground truth.

We compute the ROC curves for all the models using the mean airway completeness and volume leakage measured on the test set, and varying the threshold in the output probability maps used to obtain the final binary segmentations. Further, we compute the operating point with a fixed level of volume leakage (13%) by estimating the correct threshold (with a minimum error 10^{-4}). This threshold is, for the different models: UNetLev3 (0.1), UNetLev5 (0.04), UGnnReg (0.66), UGnnDyn (0.33). We use the resulting airway segmentations to evaluate the performance measures on the test set. To compare the models, we use the two-sided T-test on two related distributions.

The calculation of the dynamic graph adjacency as described in Sect. 2.2 is computationally expensive, both in memory and run time. In order to fit the model in GPU and reduce the computational cost, the searching space of candidate nodes to compute the node connectivity is constrained to a cube area of at most 5 voxels away from the given node. This limits the range for direct node connections, however long range information is still accessed via the four stacked graph convolutions as described in Sect. 2.1.

4 Results and Discussion

The ROC curves for the different models are shown in Fig. 2. The proposed UNet-GNN models and the baseline UNetLev3 show very similar results. In the detailed view, the UGnnReg shows a small improvement, i.e. higher airway completeness for a given volume leakage, over the baseline UNetLev3. Further, the UNetLev5 shows lower completeness than the other models for volume leakage higher than 11%, while the contrary occurs for lower leakage levels.

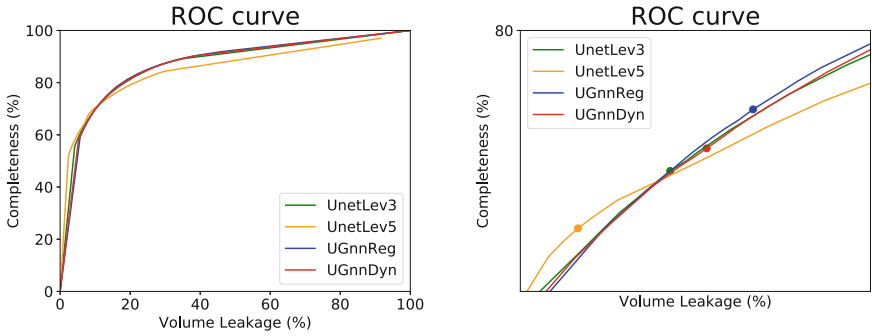


Fig. 2. ROC curves for all the models, varying the threshold in the probability maps. Right: detailed view, including the operating points with threshold 0.5.

Performance of the different models on the test set, for an operating point in the ROC curve with 13% volume leakage, is shown in Fig. 3. In dice overlap coefficient, airway completeness and volume leakage, there is no significant difference between the models ($p > 0.1$). Nevertheless, in centreline distance measures, the proposed UNet-GNN models show significantly lower d_{FN} and higher d_{FP} with respect to the baseline UNetLev3: in d_{FN} , comparing UGnnDyn ($p = 0.001$) and UGnnReg ($p < 0.01$) with UNetLev3, and in d_{FP} , comparing both UNet-GNN models with UNetLev3 ($p < 0.001$). Further, the UNetLev5 shows no significant difference in d_{FP} and d_{FN} with the other models ($p > 0.1$).

The small and significant improvement in distance false negative error d_{FN} , with an on par dice overlap and airway completeness, indicates that the proposed UNet-GNN models can segment slightly more complete airway trees,

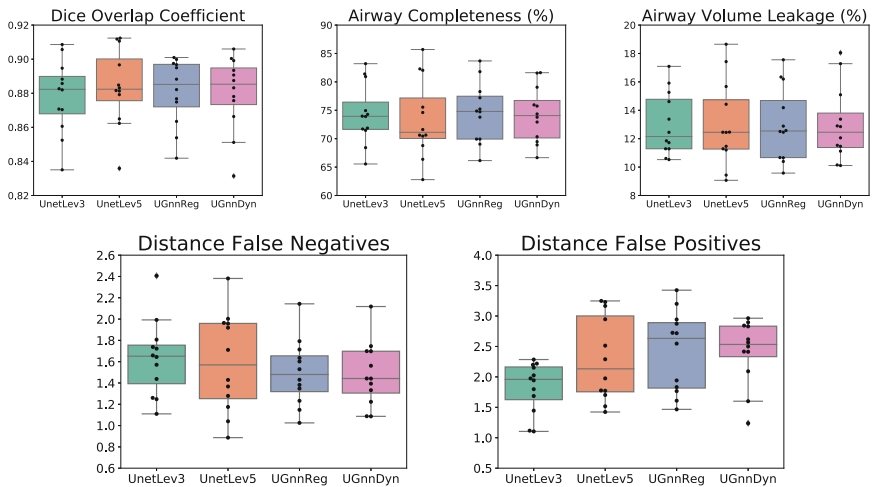


Fig. 3. Test set performance measures for the different models.

when compared to the two UNet models, with more and/or longer peripheral branches. This is because the centreline distance measure d_{FN} is not dependent on the scale of airways and provides a uniform measure of accuracy in detecting branches. In contrast, the completeness measure is a binary evaluation of branches with respect to the reference and small improvements in centreline detection may not be reflected in the overall score. On the contrary, the more complex model UNetLev5 does not show any difference in any measure over the baseline UNetLev3. Further, the UNet-GNN models have fewer trainable parameters ($\approx 50k$) than the UNetLev3 ($\approx 90k$) and much less than the UNetLev5 ($\approx 1.4M$).

One aspect that might limit the performance of the proposed UNet-GNN models is that the GNN module is operating only on the deepest level of the UNet, where the image patches have undergone three downsampling operations. A more powerful UNet-GNN model can be formulated by replacing all skip connections in the baseline UNet with GNN modules, each operating at that image resolution level. However, this model has not been experimented with due to the large memory footprint required, which exceeded the available GPU resources at our disposal.

5 Conclusions

We presented a joint UNet-GNN based segmentation method with an application to extract airways from chest CTs. By introducing a GNN module with graph convolutions at the deepest level of the UNet the proposed method is able to learn and combine information from a larger region of the image. The proposed UNet-GNN models show a small and significant improvement in the false negative centreline measure over the baseline UNet, with on par dice overlap and airway completeness, for a fixed volume leakage. This indicates that the proposed UNet-GNN models can segment slightly more complete airway trees. Further, this is achieved with fewer trainable parameters.

References

1. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
2. 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
3. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2009)
4. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
5. Lo, P., et al.: Extraction of airways from CT (EXACT’09). IEEE Trans. Med. Imaging **31**(11), 2093–2107 (2012)

6. Meng, Q., Roth, H.R., Kitasaka, T., Oda, M., Ueno, J., Mori, K.: Tracking and segmentation of the airways in chest CT using a fully convolutional network. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10434, pp. 198–207. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66185-8_23
7. Juarez, A.G.-U., Tiddens, H.A.W.M., de Bruijne, M.: Automatic airway segmentation in chest CT using convolutional neural networks. In: Stoyanov, D., et al. (eds.) RAMBO/BIA/TIA -2018. LNCS, vol. 11040, pp. 238–250. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00946-5_24
8. Selvan, R., et al.: Extraction of airways using graph neural networks. In: 1st Conference on Medical Imaging with Deep Learning, Amsterdam (2018)
9. Shin, S.Y., Lee, S., Yon, I.D., Lee, K.M.: Deep vessel segmentation by learning graphical connectivity. ArXiv preprint [arXiv:1806.02279](https://arxiv.org/abs/1806.02279) (2018)
10. Petersen, J., et al.: Optimal surface segmentation using flow lines to quantify airway abnormalities in chronic obstructive pulmonary disease. *Med. Image Anal.* **18**, 531–541 (2014)
11. Lo, P., Sporning, J., Ashraf, H., Pedersen, J.H., de Bruijne, M.: Vessel-guided airway tree segmentation: a voxel classification approach. *Med. Image Anal.* **14**(4), 527–538 (2010)
12. Lo, P., Sporning, J., Pedersen, J.J.H., de Bruijne, M.: Airway tree extraction with locally optimal paths. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) MICCAI 2009. LNCS, vol. 5762, pp. 51–58. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04271-3_7
13. Pedersen, J., Jesper, H., et al.: The Danish randomized lung cancer CT screening trial overall design and results of the prevalence round. *J. Thorac. Oncol.* **4**(5), 608–614 (2009)