ABSTRACT

We propose a model for the joint segmentation of the liver and liver lesions in computed tomography (CT) volumes. We build the model from two fully convolutional networks connected in tandem and trained together end-to-end. The first network is trained to produce a representation that is used for liver segmentation. This representation is passed to every layer in the second network, the output of which is used to produce a lesion segmentation. We evaluate the approach on the 2017 ISBI Liver Tumour Segmentation Challenge and place second with a per-volume average Dice score of 0.65.

Index Terms—segmentation, neural network

1. INTRODUCTION

The segmentation of malignant tumours in computed tomography (CT) is important for cancer diagnosis, treatment planning, and tracking treatment response. Manual segmentation is time consuming and is often omitted in lieu of an inaccurate volume estimate from the two largest diameters of the tumour in the transverse plane [1]. Nevertheless, tumour volume is a better predictor of patient survival than diameter [2]. There is a clear need for tools to aid with tumour detection and segmentation.

Recent advances in computer vision have spurred the resurgence and refinement of deep neural networks which can now exceed human performance in object classification from natural images [3]. Exploration of this promising avenue has only recently begun for medical image segmentation. Current models [4,5,6,7,8] are based on fully convolutional neural networks (FCN) such as the one by Long et al. [9] or Hariharan et al. [10] and especially on the UNet [11]. We exploit the architecture that is evaluated in [4] to construct a model configuration for segmenting metastatic lesions in the liver within CT volumes. Our approach placed second in the 2017 ISBI Liver Tumour Segmentation Challenge (LiTS).

2. DATASET AND PREPROCESSING

The proposed segmentation method was applied to metastatic lesions in the liver imaged with CT. The dataset was comprised of 200 CT volumes, each containing the liver but spanning variable focii, from abdominal to abdominal and thoracic. All volumes were enhanced with a contrast agent, imaged in the portal venous phase. All volumes contained a variable number of axial slices with a resolution of 512x512 pixels, with varying slice thicknesses. Of the 200 volumes, 130 volumes were provided publicly with manual segmentations of the liver and liver lesions while 70 were withheld until near the end of the LiTS challenge for evaluation. Manual segmentations were not provided for this evaluation set.

Of the 130 cases with segmentations, we used 100 for training and 15 for validating our segmentation models. We did not use the 15 which lacked lesion segmentations (though we intend to do so later). We did not apply any preprocessing to the images except for basic image-independent scaling of the intensities to ensure inputs to our neural networks were within a reasonable range: we divided all pixel values by 255 and then clipped the resulting intensities to within [-2, 2].

3. METHOD

We assembled a model composed of two FCNs, one on top of the other, trained end-to-end to segment 2D axial slices. Both networks are UNet-like with short and long skip connections as in [4]. The combined network is shown in Figure 1. FCN 1 takes an axial slice as input and its output is passed to a linear classifier that outputs (via a sigmoid) a probability for each pixel being within the liver. FCN 2 takes as input both the axial slice and the output of FCN 1. The input thus has a number of channels equal to the number of channels in the representation produced by FCN 1 plus one channel which contains the axial slice. The output of FCN 2 is passed to a lesion classifier, of the same type as the liver classifier.

This configuration allows FCN 2 to benefit from a liver segmentation when performing lesion segmentation, allow-
The overall structure of the segmentation model. Two FCNs, FCN 1 and 2, each take a 2D axial slice as input. FCN 2 additionally takes the representation produced by FCN 1 as input. While FCN 1 is used to produce a liver segmentation, FCN 2 is used to produce a lesion segmentation.

Alternatively, one could separately segment the liver and then manually limit lesion segmentation to within the liver, but this precludes the flexibility afforded to FCN 2 for determining how to limit its segmentation. This flexibility may be important because the liver segmentation is imperfect and always cropping lesion segmentations to the liver could result in correct parts of lesions being cropped out. In addition, training FCN 1 and 2 end-to-end allows FCN 1 to learn a representation that is amenable for lesion segmentation, boosting the performance of FCN 2. The presence of liver pixel labels in the dataset allows us to include this initial FCN 1 network without suffering overfitting from an increase in the number of parameters compared to just using FCN 2 in isolation.

The FCN 1 and 2 networks have an identical architecture, as shown in Figure 2 at the top. In each FCN, an input passes through an initial convolution layer and is then processed by a sequence of convolution blocks at decreasing resolutions and an increasing receptive field size. This contracting path is shown in blue on the left. An expanding path (right, in yellow) then reverses the downsampling performed by the contracting path. The expanding path mirrors the structure of the contracting path. Each block in the expanding path takes as input the sum of the previous block’s output and the output of its corresponding block from the contracting path; this allows the expanding path to recover spatial detail lost with downsampling. Representations are thus skipped from left to right along long skip connections.

We used two types of blocks: block A and block B. Both have short skip connections which sum the block’s input into its output, as shown in Figure 2 (bottom). Both blocks contain dropout layers, a downsampling layer when used along the contracting path, and an upsampling layer when used along the expanding path. The downsampling layer in block A (Figure 2 bottom left) is max pooling while in block B (Figure 2 bottom right) is basic grid subsampling, achieved by applying convolutions with a stride of 2. The upsampling layer performs simple nearest neighbour interpolation. The main difference between blocks A and B is in the number of convolution operations: block A contains one convolution layer and block B contains 2. All convolution layers use 3x3 filters.
filters; the number of filters is shown for each layer/block in Figure 2. Each convolution layer is preceded by a rectified linear unit (ReLU) [12] and batch normalization [3].

Because the FCNs both have short skip connections, the representation passed from FCN 1 to FCN 2 is effectively passed directly to every level of representation in FCN 2, after first being pre-processed by the first convolution layer of FCN 2.

We trained the model only on 2D axial slices that contain the liver, using RMSprop [13] and the Dice loss defined in [6, 4]. For data augmentation, we applied random horizontal and vertical flips, rotations up to 15 degrees, zooming in and out up to 10%, and elastic deformations as described by [11]. In order to improve training time, allowing us to test many models and hyperparameters in a short time, we first down-scaled all slices from a 512x512 resolution to 256x256. This initial model was trained with a 0.001 learning rate (0.9 momentum). The second stage of training was performed on full resolution slices, fine-tuning the model with a 0.0001 learning rate. Applying the model from the first stage to higher resolution slices makes sense for most lesions since they appear at vastly different scales—this is similar to simply doubling the scale of lesions. Surprisingly, it works well for the liver as well, which maintains a roughly consistent size.

We also tried adding an additional block (type A) to the contracting and expanding paths to account for the change in input scale. In that case, we initially froze the weights trained in stage 1, re-initialized the first and last convolution layer, and trained them together with the new blocks; then, we unfroze all weights and fine-tuned the network. This did not perform better than the approach described above.

We used Titan X (GM200, Maxwell) GPUs for model training and inference. The model was trained for 250 epochs on downsampled slices (batch size 40, 730s per epoch) and fine-tuned for 30 epochs on full-resolution slices (batch size 10, 3000s per epoch). In both stages of training, the weights which yielded the best loss on the validation set were kept. At test time, the network produced 15 full-resolution segmentation predictions per second.

At test time, segmentation predictions were averaged across all four input orientations achieved by vertical and horizontal flips. These were then further averaged across three similar models. A liver segmentation was extracted by selecting the largest connected component in the model’s liver segmentation. A lesion segmentation was extracted by cropping the model’s lesion segmentation to a dilated version of the liver segmentation. For dilation, we chose to extend the liver’s boundaries by 20mm. This eliminated most false positives outside of the liver without incorrectly cropping out lesions when the liver is slightly under-segmented. No further post-processing was performed on the model outputs.

The proposed method scored an average per-volume Dice of 0.65 on the LiTS challenge, placing second in ranking. Example segmentations are shown in Figure 3. The top six examples present fairly good segmentation results while the bottom four examples are of failed segmentations. Lesion segmentations successfully produced across the full range of lesion sizes. The model struggled most with tiny nodular lesions (comprised of just a few voxels). Failure examples show missed nodular lesions and a missed lesion segmentation due to a poorly segmented liver. Overall, liver segmentation performed well, with an average per-volume Dice score of 0.95 on the validation set. Lesion segmentation performed poorly on the small validation set, with a dice of only 0.52.

The proposed model is limited to processing 2D slices. This restriction is only due to memory constraints. We expect that incorporating cross-slice context into the model would significantly improve segmentation results. This can be done for example by adding a transition between representations across slices (trained with truncated backpropagation through time) or, once GPUs have enough memory, by simply using 3D convolutions instead of 2D convolutions.

We chose to separate FCN 1 and FCN 2 instead of using a single FCN for both liver and lesion segmentation. Doing the latter results in poor segmentation performance. This may be in part because lesion segmentation cannot then benefit as much from attention to the liver. The liver representation evolves across the entire network and is not complete until the network’s output. It is also not captured in full scale until the coarsest level of representation (in the middle of the network). The FCN structure can be seen in Figure 2 for reference. The coarsest representation is at the bottom (middle) layer of the network. By cascading FCN 1 and FCN 2, FCN 2 benefits from a complete liver representation produced by FCN 1.

It is also possible that FCN 1 helps to standardize the intensity distribution of the inputs to FCN 2 as suggested by [14]. We found that there were some cases with incorrectly normalized intensities where the Hausdorff units deviated by up to three orders of magnitude from expected values; these were mostly handled well by the combined network. Although FCN 2 includes the same raw axial slice in its input as FCN 1, the additional information passed from FCN 1 could serve as a point of reference for standardizing the inputs to FCN 1. It would be interesting to evaluate, for varying input intensity scales, the change in the statistics of the representation after the first convolution layer of FCN 2 (“in conv” in Figure 1 “conv 3x3” in Figure 2).

The proposed model performs joint liver and lesion segmentation quickly without any need for preprocessing of input images or complicated postprocessing of the outputs. Furthermore, although it is limited to processing 2D axial slices independently, this restriction can be easily overcome. Thus, the proposed model could serve as a good base model which...
can be further improved for liver and liver lesion segmentation.

5. REFERENCES

[1] EA1 Eisenhauer, Patrick Therasse, Jan Bogaerts, LH Schwartz, D Sargent, Robert Ford, J Dancey, S Arbuck, S Gwyther, M Mooney, et al., “New response evaluation criteria in solid tumours: revised recist guideline (version 1.1),” European journal of cancer, vol. 45, no. 2, pp. 228–247, 2009.

[2] Julius Chapiro, Rafael Duran, MingDe Lin, Rüdiger E Schernthaner, Zhijun Wang, Boris Gorodetski, and Jean-François Geschwind, “Identifying staging markers for hepatocellular carcinoma before transarterial chemoembolization: comparison of three-dimensional quantitative versus non-three-dimensional imaging markers,” Radiology, vol. 275, no. 2, pp. 438–447, 2014.

[3] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” arXiv preprint arXiv:1502.03167, 2015.
[4] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal, “The importance of skip connections in biomedical image segmentation,” in International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis. Springer, 2016, pp. 179–187.

[5] Patrick Ferdinand Christ, Mohamed Ezzeldin A Elshaer, Florian Ettlinger, Sunil Tatavarty, Marc Bickel, Patrick Bilic, Markus Rempfler, Marco Armbruster, Felix Hoffmann, Melvin DAnastasi, et al., “Automatic liver and lesion segmentation in ct using cascaded fully convolutional neural networks and 3d conditional random fields,” in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2016, pp. 415–423.

[6] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” in 3D Vision (3DV), 2016 Fourth International Conference on. IEEE, 2016, pp. 565–571.

[7] Rudra PK Poudel, Pablo Lamata, and Giovanni Montana, “Recurrent fully convolutional neural networks for multi-slice mri cardiac segmentation,” arXiv preprint arXiv:1608.03974, 2016.

[8] Hao Chen, Xiaojuan Qi, Jie-Zhi Cheng, and Pheng-Ann Heng, “Deep contextual networks for neuronal structure segmentation,” in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press, 2016, pp. 1167–1173.

[9] Jonathan Long, Evan Shelhamer, and Trevor Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[10] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik, “Hypercolumns for object segmentation and fine-grained localization,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 447–456.

[11] 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. Springer, 2015, pp. 234–241.

[12] Vinod Nair and Geoffrey E Hinton, “Rectified linear units improve restricted boltzmann machines,” in Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814.

[13] T. Tieleman and G. Hinton, “Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude,” COURSERA: Neural Networks for Machine Learning, 2012.

[14] Michal Drozdzal, Gabriel Chartrand, Eugene Vorontsov, Lisa Di Jorio, An Tang, Adriana Romero, Yoshua Bengio, Chris Pal, and Samuel Kadoury, “Learning normalized inputs for iterative estimation in medical image segmentation,” arXiv preprint arXiv:1702.05174, 2017.