2D-Densely Connected Convolution Neural Networks for automatic Liver and Tumor Segmentation

Krishna Chaitanya Kaluva, Mahendra Khened, Avinash Kori and Ganapathy Krishnamurthi

Abstract—In this paper we propose a fully automatic 2-stage cascaded approach for segmentation of liver and its tumors in CT (Computed Tomography) images using densely connected fully convolutional neural network (DenseNet). We independently train liver and tumor segmentation models and cascade them for a combined segmentation of the liver and its tumor. The first stage involves segmentation of liver and the second stage uses the first stage’s segmentation results for localization of liver and henceforth tumor segmentations inside liver region. The liver model was trained on the down-sampled axial slices $(256 \times 256)$, whereas for the tumor model no down-sampling of slices was done, but instead it was trained on the CT axial slices windowed at three different Hounsfield (HU) levels. On the test set our model achieved a global dice score of 0.923 and 0.625 on liver and tumor respectively. The computed tumor burden had an rmse of 0.044.

Keywords—CNN, FCN, DenseNet, Liver Tumor, dice loss.

I. INTRODUCTION AND RELATED WORK

The exponential growth in the number of medical images of different modalities used in clinical practice has led to an interest in developing automated methods for analysis of medical images. Computed tomography (CT scan) is a noninvasive diagnostic imaging procedure that uses a combination of X-rays and computer algorithms to generate different internal maps created at a given resolution. A denseblock refers to concatenation of new feature maps which is accomplished by using a $1 \times 1$ convolution (depth preserving) followed by a $2 \times 2$ max-pooling operation. A denseblock refers to concatenation of new feature maps created at a given resolution.

II. OUR METHOD

CT Windowing is a technique frequently used in the evaluation of CT scans for the purpose of enhancing contrast of particular type of tissue or the abnormality type being examined. The abdominal CT images of a patient comprises of various organs such as liver, spleen, gall bladder, etc. Anatomically, the HU range of liver is $60 \pm 6$. Our method is based on 2-stage cascaded approach for segmentation of liver and its tumors from HU windowed CT volumes. In this method we first train the liver model for the task of liver segmentation. Since the shape (contour) and texture of liver is simple when compared to its tumors, the CT images after windowing were down-sampled to half their original size $(512 \times 512)$ and then used for training. This helped in reducing the computation required for the training liver model and also helps in faster segmentation of liver. The tumor model was 3-channel input and was trained independently on full-sized CT images of liver windowed at 3 different levels having different window widths. During prediction, the liver segmentation from
first stage in the cascade (liver model) aids the second stage (tumor model) by precisely localizing on liver regions in CT images to produce combined predictions of liver and tumor segmentations.

A. DATASET AND PREPROCESSING

The models were trained and tested on the LITS MICCAI-2017 challenge dataset which comprised of 200 contrast-enhanced CT images taken at different phases (mostly venous phase) with only a few cases with anomalies like fatty liver, cirrhosis liver, calcification in the liver, etc. Out of 200 CT scans, for the 130 scans radiologist hand-drawn ground-truths were given for training the model and the rest 70 were used for testing by the challenge organizers. We divided the 130 training dataset into train: 90 volumes, validation: 26 volumes, test: 14 volumes.

For liver model, the following pre-processing techniques were done on the CT volumes in the order specified:
1) HU values are windowed to the range of [-100,300].
2) 0 − 1 min-max normalization on the entire volume.
3) Down-sample the slices from 512 × 512 to 256 × 256.

For tumor model, the following pre-processing techniques were done on the CT volumes in the order specified:
1) 3 different HU windowing ranges ([0,100], [-100,200], [-100,400]) were used to produce 3 images.
2) 0 − 1 min-max normalization on the entire volume.

In most of the CT volumes (in the challenge dataset), the liver and tumor slices comprised of a small fraction of the total volume. So in order to address the data imbalance, the liver model was trained only on liver slices and additional 10 slices were taken above and below the liver. Similarly, the tumor model was trained only on tumor slices and with additional 5 slices above and below the tumor.

B. Liver Model

1) Network Architecture:
Fig. 1 illustrates the schematic diagram of the proposed network for liver segmentation. Our proposed network doesn’t have skip connections. The up-sampling and down-sampling paths uses fully convolutional denseblocks. Each denseblock comprises of 4 layer blocks and each layer in the denseblock is sequentially composed of BN → ELU → 3 × 3 convolution layer. The first Dense-Block was prefixed with a layer comprising of several convolution filters of size 3 × 3 on the input images. In the down-sampling path, the input to a dense block was concatenated with its output, leading to a linear growth of the number of feature maps. The Transition-Down block (TD) consists of BN → ELU → 1 × 1 convolution and → 2 × 2 max-pooling layers. The last layer of the down-sampling path is referred to as Bottleneck.

In the up-sampling path, the input of a Dense-Block is not concatenated with its output. Transition-Up (TU) block consists of spatial-bilinear up-sampler followed by BN → ELU → and → 3 × 3 convolution layer. Our thesis is that the network also learns better way of up-sampling the outputs which resulted in better predictions. To get the final segmentation map of size of 512 × 512 a simple spatial bilinear up-sampling block (SBU) is added in the penultimate layer. The feature maps of the hindmost up-sampling component was convolved with a 3 × 3 convolution layer followed by a softmax layer to generate the final segmentation. To prevent over-fitting, a dropout rate of 0.2 was implemented following each convolution layer.

Table (I) summaries the individual blocks of network architecture. From left to right: layer used in the model, Transition Down (TD) and Transition Up (TU).

2) Loss function:
Liver being the largest organ in the body, the class imbalance shown in the CT volume is minimal, but contouring of the liver borders from its neighboring organs in CT images is generally not precise. Hence in-order to predict the edges...
precisely, weight-maps were generated from the given ground truth segmentation (see Figure 2), edges of the liver were weighed higher relative to the interior regions of the liver. These weight-maps were used during training for minimizing the spatially-weighted-cross-entropy loss function. The use of weight-mapping leads to adding a heavy penalty in cost function for predicting imprecise liver contours during training the model.

The network architecture for tumor model was similar to liver model, except that it accepted 3-channel inputs and there was no simple spatial bilinear up-sampling block (SBU) as in the liver model. The 3 channels were feed with the same slice of the CT volume, but each channels HU - windowing was at 3 different levels having different window widths. It was observed that by providing 3 different HU windowed channels, the model is able to learn the tumor and its boundary very well compared to a single channel input. To prevent over-fitting a dropout rate of 0.2 was implemented.

2) Loss function:
The shape of tumors is diffusive, inhomogeneous and sparsely located in the CT volumes. This leads to class imbalance in the dataset, thereby making it hard for the network to learn the intricate features of the tumor. Hence in order to reduce the class imbalance we employ two strategies:
- Use of weight-maps with tumor portion being weighed very high compared to background. The weight-maps are used for calculation of spatially-weighted-cross-entropy loss.
- Use weighted combination of two loss functions: spatially-weighted-cross-entropy loss and a loss function based on dice overlap co-efficient.

The dice co-efficient is an overlap metric used for assessing the quality of segmentation maps. The dice coefficient between two binary volumes can be written as:

$$DICE = \frac{2 \sum_{i} p_i g_i}{\sum_{i} p_i^2 + \sum_{i} g_i^2}$$

where the sums run over the \(N\) voxels, of the predicted binary segmentation volume \(p_i \in P\) and the ground truth binary volume \(g_i \in G\).

The parameters of the network were optimized so as to minimize the total loss, Equation. (4).

$$total\_loss = \lambda (spatially\_weighted\_cross\_entropy\_loss) + \gamma (1 - dice\_loss) + L2\_loss$$

where \(\lambda\) and \(\gamma\) are empirically assigned weights to individual losses. During training it was observed that the dice loss allowed higher overlap scores than when trained with the loss function based on the cross entropy loss alone. In this work we set \(\gamma = 0.5\) and \(\lambda = 0.5\).

3) Post processing:
The predictions of the tumor network were masked with the background taken from the liver prediction, hence the final tumor predictions were only inside the liver.

### III. Conclusion

With the pre-processing, network architecture and post processing steps discussed in this paper we were able to achieve an average liver dice of 0.93 on the 14 test volumes (mentioned in the dataset split up in the beginning) and an average tumor dice of 0.45 on the 14 test volumes. Our liver prediction have
a smoother surface (3D volume) and precise edges compared to the liver predictions from a U-net whose predictions have ridges on the surface that arise because of very finer slice wise predictions that result because of the skip connections. The true positives of tumor prediction have almost more than 50% overlap with the ground truth but the model predicts a lot of false positives like gallbladder edges, diaphragm, vessels etc. The results of our proposed 2-stage cascaded model’s predictions on the 70 CT test volumes is summarized in Tables II & III.

| Metrics                  | Computed LIVER SEGMENTATION metrics | Computed LESION SEGMENTATION metrics |
|--------------------------|-------------------------------------|--------------------------------------|
| voe                      | 0.150                               | 0.411                                |
| dice global              | 0.927                               | 0.625                                |
| dice                     | 0.912                               | 0.725                                |
| rmsd                     | 9.682                               | 2.070                                |
| rdi                      | -0.008                              | 19.705                               |
| assd                     | 6.365                               | 1.441                                |
| jaccard                  | 0.850                               | 0.589                                |
| dice_per_case            | 0.912                               | 0.492                                |
| msd                      | 45.928                              | 7.515                                |

TABLE II: Results of segmentation metrics on the test set comprising of 70 CT volumes

| Computed LESION DETECTION metrics | Computed TUMOR BURDEN          |
|-----------------------------------|--------------------------------|
| recall: 0.348                     | rmse: 0.044                    |
| precision_greater_zero: 0.211     | max: 0.194                     |
| precision: 0.117                 |                                |
| recall_greater_zero: 0.628        |                                |

TABLE III: Results of lesion detection and tumor burden estimation metrics on the test set comprising of 70 CT volumes

APPENDIX A
SEGMENTATION RESULTS
REFERENCES
[1] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.
[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
[4] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[5] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.
[6] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." arXiv preprint arXiv:1505.04597 (2015).
[7] Ciresan, Dan, et al. "Deep neural networks segment neuronal membranes in electron microscopy images." Advances in neural information processing systems. 2012.
[8] Menze, Bjoern H., et al. "The multimodal brain tumor image segmentation benchmark (BRATS)." IEEE transactions on medical imaging 34.10 (2015): 1993-2024.
[9] Jgou, Simon, et al. "The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation." arXiv preprint arXiv:1611.09326 (2016).
[10] Huang, Gao, et al. "Densely connected convolutional networks." arXiv preprint arXiv:1608.06993 (2016).
[11] Christ, Patrick Ferdinand, et al. "Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields." International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer International Publishing, 2016.
[12] Martin Abadi, et al. "TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org".
[13] Sfian van der Walt, Johannes L. Schnberger, Juan Nunez-Iglesias, Francois Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Goulart, Tony Yu and the scikit-image contributors. "scikit-image: Image processing in Python. PeerJ 2:e453 (2014)"
[14] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
[15] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research 15.1 (2014): 1929-1958.
[16] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." International Conference on Machine Learning. 2015.
[17] Clevert, Djork-Arn, Thomas Unterthiner, and Sepp Hochreiter. "Fast and accurate deep network learning by exponential linear units (elus)." arXiv preprint arXiv:1511.07289 (2015).