Kidney and tumor segmentation using an ensemble of deep neural networks

Yu Wu[1], Yu Gan[1],

Yuhang Wu[1], and Zhang Yi[*]

[1] First author, Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu 610065, PR China
[*]Corresponding author, E-mail address: zhangyi@scu.edu.cn (Z. Yi)

Abstract. For the segmentation of kidney and tumor task, we propose a two stages model that consists of several classification networks and segmentation models. The first stage is the foreground and background classification subnetwork, this stage is to recognize whether there are kidneys or tumors on images, so we propose a classification model called RD-Net which can effectively reduce the errors caused by a large of background images and improve the efficiency of the whole segmentation results. The second stage is the segmentation model used to predict the contour of the target (kidney or tumor). Therefore, we propose Att-ResUnet model and multi-scale ensemble of postprocessing methods used to integrate the predicted results of multiple models, so as to improve the accuracy of prediction results.

Keywords: Classification, Segmentation, Postprocessing.

1 The First Stage Method (The Classification Subnetwork)

At the first stage, we use several well-known classification models to recognize whether there are kidneys or tumors on each image. To further enhance the performance, we concatenate the last feature map layer of ResNet and DenseNet into an end-to-end model using backpropagation algorithm. Moreover, we ensemble classification models and work with postprocessing methods.

1.1 The Classification Model

1.1.1 ResNet and DenseNet. ResNet and DenseNet have some similarities [1]. They both have skipping connections. The difference between them is that each layer in ResNet can have at most one skip connection but DenseNet has skipping connections from any layer to every subsequent layer of it [2].

The most significant contribution of ResNet is the proposition and application of residual connection. Residual connection solves the vanishing gradient problem so that it can make neural network be very deep [3]. A building block with residual connection, called residual block, is defined as:

$$y = f(f(W_l \cdot x) \cdot W_{l+1} + x) \tag{1.1}$$

where *y* and *x* are input and output vectors of the residual block, W_l and W_{l+1} mean the weight of stacked $layer_l$ and $layer_{l+1}$, and $f(\cdot)$ represents the activation function—ReLU [4].

The way of DenseNet using skipping connection is called dense connection. The input x_l and output y_l of layer l can be defined as:

$$x_l = h(y_0, y_1, ..., y_{l-1}) \tag{1.2}$$

$$y_l = f(W_l \cdot x_l) \tag{1.3}$$

where $(y_0, y_1, ..., y_{l-1})$ is the result that comes from the concatenation of all the feature maps of $layer_0$, $layer_l$,…, $layer_{l-1}$, $h(\cdot)$ means concatenation, W_l refers the weight of $layer_l$, and $f(\cdot)$ represents the activation function—ReLU.

1.1.2 RD-Net. Though ResNet and DenseNet are outstanding models, they work individually in most work with some limitations. In this study, we use a brand-new network architecture which consists of ResNet and DenseNet, and we call it RD-Net. So to enhance the learning of the model, we combine the features of ResNet and DenseNet into a new network to recognize the kidneys and tumors. We adopt all layers except *fc* and *softmax* layers in ResNet and all layers except *fully-connected* and *softmax* layers in DenseNet. Then we concatenate the outputs of average pool layer in ResNet and DenseNet as the input of a fully-connected layer. And next through a *softmax* layer, RD-Net gives its result of classification. Assuming that p_1 and p_2 represent the outputs of average pool layer in ResNet and DenseNet respectively, the output *y* of fully-connected layer in RD-Net can be defined as:

$$y = f(W \cdot h(p_1, p_2)) \tag{1.4}$$

Where $h(\cdot)$ means concatenation, *W* refers the weight of $layer_l$, and $f(\cdot)$ represents the activation function—ReLU.

Compared with individual ResNet and DenseNet, RD-Net retains the residual blocks and dense blocks, absorbs the feature maps of ResNet and DenseNet at last, and propagates same error back to ResNet part and DenseNet part. Maybe propagating of the same error is the reason that why RD-Net has better performance in classification task than other models on validation set.

In this study, we choose ResNet152 and DenseNet121 to build RD-Net, and both of them are pretrained from ImageNet. The architecture of RD-Net is shown in Fig. 1. Firstly, we get the feature maps from ResNet152 and DenseNet121 which has 512 dimensions and 1024 dimensions respectively. And then we concatenate these

two feature maps into a new feature map that has 1563 dimensions. Finally, we use a [1536×2] fully-connected layer and a softmax layer to carry out the classification task. To recognize whether there are kidneys and tumors on images, we put images in RD-Net to screen out which contain kidneys, then we use these images as inputs to RD-Net to recognize whether there are tumors on them.

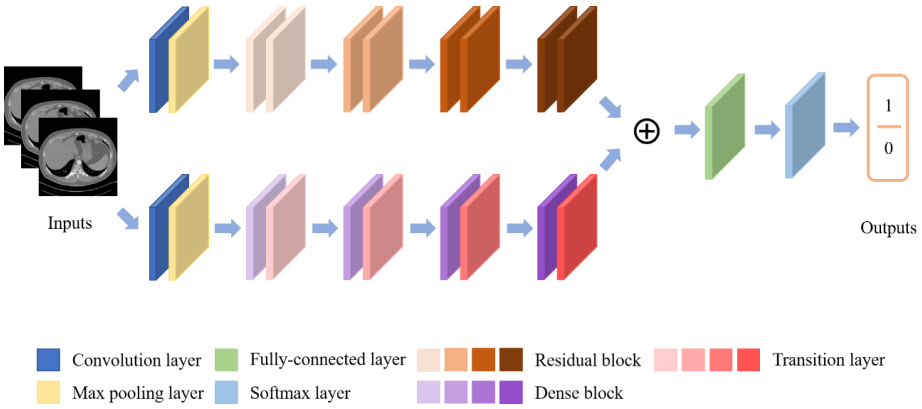

Fig. 1. The proposed architecture of RD-Net.

1.2 Ensemble of Classification models

To further improve the accuracy of classification, we ensemble the results that obtained from ResNet, DenseNet and RD-Net through voting method. Classification result of each image is decided by more than half of models that have same predictions. Ensemble operation considers predictions of all models and gets a usual better performance [5].

1.3 Classification Postprocessing

We know that the CT images in each case are sequences. So the images which have kidneys or tumors must be contiguous. It is impossible that there are some images which don't have kidneys or tumors while their previous images and subsequent images have kidneys or tumors, or there are some images which have kidneys or tumors while their previous images and subsequent images don' have kidneys or tumors. For recognition task of kidneys, we assume that *1* means there are kidneys on image, *0* means there are no kidneys on image and *L* means the outputs list of RD-Net in one case. So *L* might be a list like {0,0,0,1,0,0,…,1,1,1,0,1,1,1,…,0,0,0,} corresponding to the ordered images of the case. At first, we give all the element a new property called "believe", and set all the "believe" to 0.

And the following Fig. 2 is the algorithm of postprocessing.

```
for i← 0 to length[L]:
  if L[i].value = 1 and i not equal to 0:
    if not (L[i-1].believe = 0 and L[i+1].value = 0):
      L[i].believe ← 1
  else if L[i] = 0:
    L-nexts ← L[i+1] to L[i+1+n]
    if L[i-1].believe = 1 and any(L-nexts) = 1:
      L[i].believe ← 1
  rerurn L.believe
end
```

Fig. 2. The algorithm of classification postprocessing.

After the postprocessing, we get a new list *L-new*. *L-new* is just like {0,0,0,0,0,0,…,1,1,1,1,1,1,1,…,0,0,0,}. Images corresponding to *1* are what we adopt to put into the second stage.

2 The Second Stage Method (The Segmentation Task)

The segmentation task is the critical part used to implement the semantic segmentation work of CT datasets. Our model consists of two main parts: a series of DeepLab models and a variant of Unet, called Att-ResUnet.

2.1 The Segmentation Network Model

2.1.1 DeepLab. DeepLabv3 is a recent verson which is a powerful tool that adjusts the view of filter field and controls the feature response resolution of convolutional neural network computation in semantic segmentation tasks. In order to solve the problem of multi-scale target segmentation, it also designs the atrous convolution in cascade or in parallel architecture with different sampling rates [6].

Taking 2-dimensional signals for example, for every position a of the feature map y and the convolution filter v, when inputting feature map x, the atrous convolution is:

$$y[a] = \sum_{k=0}^{n} x[a + r * k]v[k]\binom{n}{k} \tag{2.1}$$

And the atrous rate r is connected with the stride which featuring the input signal.

On the basis of DeepLabv3, DeepLabv3+ adds a simple and effective decoding module to improve the segmentation effect, especially for the boundary of object. Based on the proposed encoder-decoder structure, it can arbitrarily control atrous convolution to output the resolution of encoding features, balance the precision and running time [7].

2.1.2 Att-ResUnet. Unet is a modified network based on FCNs [8], which consists of two parts: the downsampling layers for extracting features and the upsampling layers for fusing features [9]. It's a suitable model for medical image segmentation, requiring very few datas to complete end-to-end training and achieving excellent results.

Inspired by DenseUnet [10], we use the residual module of ResNet to replace the module of Unet. However, since ResNet has more parameters, we add the attention mechanism to the upsampling part, which can prevent the parts from learning the unrelated features in the model and at the same time stress the model to learn features related to tasks [11]. So we choose a Unet backbone architecture which is the encoder-decoder framework for more information on the original image textures spreading in high resolution layers [12]. In order to reach continuous learning as the number of model layers increases, the layers of Unet framework are replaced by improved residual blocks of convolutional network [13]. Residual blocks effectively reduce the problems of gradient disappearance and explosion in deep networks [14]. As a consequence, we name the model Att-ResUnet because it contains the modified residual blocks with varied atrous convolutions and a Unet framework with attention mechanism.

To compute the attention vector at each output time t over the input words ($1,\ldots,TA$) we define:

$$u^t = v^T \tanh(W_1 h_i + W_2 d_t) \tag{2.2}$$

$$a^t = softmax(u^t) \tag{2.3}$$

$$d^t = \sum_{i=0}^{T} a^t h_i \tag{2.4}$$

The vector v and matrices W_1, W_2 are learnable parameters of the model. The vector u^t has length T_A and its *i-th* item contains a score of how much attention should be put on the *i-th* hidden encoder state h_i. These scores are normalized by softmax to create the attention mask at over encoder hidden states. In all our experiments, we use the same hidden dimensionality at the encoder and the decoder, so v is a vector and W_1 and W_2 are square matrices. Lastly, we concatenate d_t which becomes the new hidden state from which we make predictions, and which is fed to the next time step in our recurrent model [15].

Firstly, we choose Resnet152 as the basic downsampling block in the framework's encoder part [16]. In this model part, the output of each of the residual blocks is downsampled with a convolution of kernel size of one and stride of one. And then in the decoder part, the upsampling is being done with the use of attention layer followed by a ReLU activation function and a normal convolution with a kernel size of one [17]. The combination of layers from the encoder and decoder parts is also being achieved with the attention layer [18]. This layer concatenates the two inputs and subjects them to a normal convolution which brings the number of features to the requisite size.

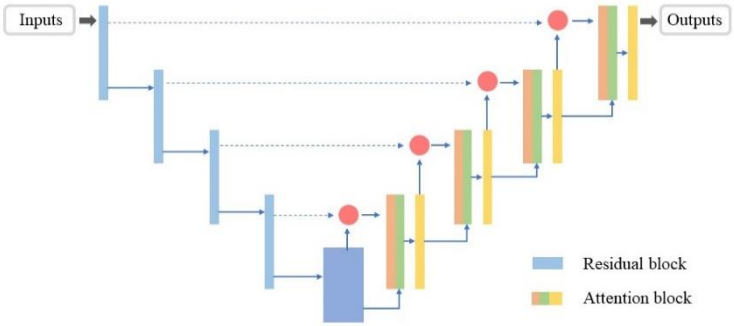

Fig. 3. The proposed architecture of Att-ResUnet.

2.2 Multi-scale and Segmentation Ensemble

We resize images in the test set into 10 different sizes, which are {300,350,420,450,512,620,650,720,750,800}. It is equivalent that we have 10 different sizes of test sets. All images in the same test set have same size which is different from the sizes of images in other test sets. Finally, we get 10 dice scores corresponding to 10 test sets.

According to scores that got from multi-scale operation, we choose the prediction sets of Top5 scores corresponding sizes to ensemble. First of all, we resize predictions into original size (512×512), so that for each original image there are 5 different predictions. Next, we generate a new prediction image with all the pixel values of 0. For one original image, we traverse all the pixels in its 5 predictions. If more than two predictions have same pixel value at the same position, we set the pixel at this position on the new prediction image to the new value. After the ensemble operation [19], final version of segmentation results is generated.

2.3 The Connected Component

The Connected Component processing is one of the post-processing methods we use to clean up the mask images generated by our segmentation network models. The method can be used in application scenarios where foreground targets need to be extracted for subsequent processing, and usually the object of connected component processing is a binary image. Generally, it refers to the image region composed of foreground pixels with the same pixel value and adjacent positions in the image [20]. Description of the algorithm is as following:

for each row (or column) in a binary image:

1. record the starting and ending positions of each sequence of pixels we choose for this row;
2. except for the first row, determine whether there is any overlap with the previous row sequence;

If there is no overlap:
allocating a new tag;
if there is one overlap:
marking with the tags from the previous sequence;
if there is more than one overlap:
marking with the smallest one in a row of overlapping sequences. Meanwhile, the following tags are denoted as equivalence pairs with this tag.

end

References

1. He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." ArXiv.org (2015): ArXiv.org, Dec 10, 2015.
2. Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely Connected Convolutional Networks." (2016).
3. Li, Hao, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. "Visualizing the Loss Landscape of Neural Nets." (2017).
4. Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks." (2011): 315-23.
5. Zhou, Zhi-Hua. Ensemble Methods: Foundations and Algorithms. Chapman & Hall, 2012. Chapman & Hall/CRC Data Mining and Knowledge Discovery Ser.
6. Chen, Liang-Chieh, George Papandreou, Florian Schroff, and Adam Hartwig. "Rethinking Atrous Convolution for Semantic Image Segmentation." ArXiv.org (2017): ArXiv.org, Dec 5, 2017.
7. Chen, Liang-Chieh, Yukun Zhu, George Papandreou, Florian Schroff, and Adam Hartwig. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation." ArXiv.org (2018): ArXiv.org, Aug 22, 2018.
8. Shelhamer, Evan, Jonathan Long, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence 39.4 (2017): 640-51. Web.
9. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." ArXiv.org (2015): ArXiv.org, May 18, 2015.
10. Guan, Steven, Amir Khan, Siddhartha Sikdar, and Parag Chitnis. "Fully Dense UNet for 2D Sparse Photoacoustic Tomography Artifact Removal." IEEE Journal of Biomedical and Health Informatics PP.99 (2019): 1.
11. Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective Approaches to Attention-based Neural Machine Translation." (2015).
12. Oktay, Ozan, Jo Schlemper, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. "Attention U-Net: Learning Where to Look for the Pancreas." ArXiv.org (2018): ArXiv.org, May 20, 2018.
13. Gu, Zaiwang, Jun Cheng, Huazhu Fu, Kang Zhou, Huaying Hao, Yitian Zhao, Tianyang Zhang, Shenghua Gao, and Jiang Liu. "CE-Net: Context Encoder Network for 2D Medical Image Segmentation." IEEE Transactions on Medical Imaging PP.99 (2019): 1.
14. Zhang, Zhengxin, Qingjie Liu, and Yunhong Wang. "Road Extraction by Deep Residual U-Net." IEEE Geoscience and Remote Sensing Letters 15.5 (2018): 749-53.

15. Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate." ArXiv.org (2016): ArXiv.org, May 19, 2016.
16. Zeiler, M D, D. Krishnan, G W Taylor, and R. Fergus. "Deconvolutional Networks." 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010): 2528-535.
17. Zeiler, Matthew, and Rob Fergus. "Visualizing and Understanding Convolutional Networks." ArXiv.org (2013): ArXiv.org, Nov 28, 2013.
18. Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." ArXiv.org (2016): ArXiv.org, Oct 10, 2016.
19. Maji, Debapriya, Anirban Santara, Pabitra Mitra, and Debdoot Sheet. "Ensemble of Deep Convolutional Neural Networks for Learning to Detect Retinal Vessels in Fundus Images." ArXiv.org (2016): ArXiv.org, Mar 15, 2016.
20. Grana, Costantino, Federico Bolelli, Lorenzo Baraldi, and Roberto Vezzani. "YACCLAB - Yet Another Connected Components Labeling Benchmark." 2016 23rd International Conference on Pattern Recognition (ICPR) (2016): 3109-114.