Segmenting Kidney on Multiple Phase CT Images using ULBNet

Yanling Chi[1*] , Yuyu Xu[2*], Gang Feng[2], Jiawei Mao[3], Sihua Wu[3], Guibin Xu[2#], Weimin Huang[1#]

[1]Institute for Infocomm Research, Agency for Science, Technology and Research, 1 Fusionopolis Way #21-01 Connexis (South Tower) Singapore 138632, chiyl@i2r.a-star.edu.sg, wmhuang@i2r.a-star.edu.sg

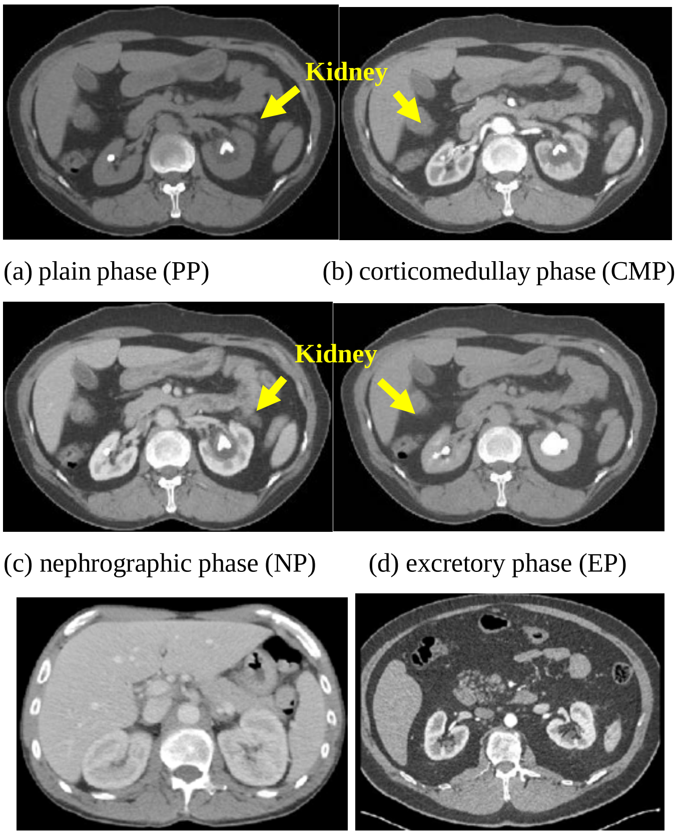
[2]Institute Department of Urology, Fifth Affiliated Hospital of Guangzhou Medical University, Guangzhou 510700, China, gyxyy@foxmail.com, gygwfg@163.com, gyxgb@163.com

[3]Creative Medtech Solutions Pte Ltd, mao.jiawei@ultrastmedtech.com, carter.wuc@163.com

[*] Equal contribution authors, [#]Correspondence authors

***Abstract*—Segmentation of kidney on CT images is critical to computer-assisted surgical planning for kidney interventional therapy. Segmenting kidney manually is impractical in clinical, automatic segmentation is desirable. U-Net has been successful in medical image segmentation and is a promising candidate for the task. However, semantic gap still exists, especially when multiple phase images or multiple center images are involved. In this paper, we proposed an ULBNet to reduce the semantic gap and to improve segmentation performance. The proposed architecture includes new skip connections of local binary convolution (LBC). We also proposed a novel strategy of fast retraining a model for a new task without manually labelling required. We evaluated the network for kidney segmentation on multiple phase CT images. ULBNet resulted in an overall accuracy of 98.0% with comparison to Resunet 97.5%. Specifically, on the plain phase CT images, 98.1% resulted from ULBNet and 97.6% from Resunet; on the corticomedullay phase images, 97.8% from ULBNet and 97.2% from Resunet; on the nephrographic phase images, 97.6% from ULBNet and 97.4% from Resunet; on the excretory phase images, 98.1% from ULBNet and 97.4% from Resunet. The proposed network architecture performs better than Resunet on generalizing to multiple phase images.**

Keywords—Kidney Segmentation; Multiple phase CT images; Network Retraining Strategy; Resunet; LBC;



(a) plain phase (PP) (b) corticomedullay phase (CMP)

(c) nephrographic phase (NP) (d) excretory phase (EP)

(e) CT images (CMP and NP) acquired from different centers.

Fig.1. Multiple phase renal CT images.

I. Introduction

Kidney segmentation on CT images is important to clinical applications for both diagnosis and treatment. As different imaging protocols are used in different clinical applications, the CT images acquired are different as in Fig. 1. For example, for characterization of renal lesions, contrast enhanced CT images are required, while non-contrast CT is acquired for planning treatment of renal stones. Sometimes, one application acquired both contrast and non-contrast images. Multiple phase CT images of the kidney generally has four phases [1]: the plain phase (PP) images acquired before contrast agent injection, the corticomedullay phase (CMP) images acquired between 25 and 80 sec after contrast agent injection, the nephrographic phase (NP) image acquired between 85-120 sec after contrast agent injection, and the excretory phase (EP) images acquired between 3-5 mins after contrast agent injection. Non-contrast enhanced CT images refers to those acquired during the plain phase. Contrast enhanced images refers to those acquired during other three phases. Convolutional neural network (CNN) has made good achievement in medical image segmentation these years. Large data and labeling are generally prerequisites in applications of deep CNN. Manually labelling is very tedious and time consuming. For a new task, to speed up the process, better practice can be making use of public available data to train the model and adapt to the specific new coming datasets. Efforts to high performance network architecture that can reduce semantic gap are desirable.

U-Net is one of most popular network architectures for medical image segmentation, first introduced by Ronneberger et al [2]. Many researchers have built their work on it and proposed novel modifications on the architecture since then.

Cicek et al extended U-Net to 3D U-Net [3]. Zhou et al proposed UNet++, which redesigned skip pathways by having convolutional layers with dense connections on the skip pathways [4]. Iglovikov et al. proposed a TernausNet, which replaced the encoder of U-Net with VGG11 weight pre-trained on ImageNet [5]. Isensee proposed a cascaded U-Net, nnU-Net, to dynamic adapt to tackle large images [6]. Milletari et al proposed V-Net which conducted 3D volumetric convolution and soft dice loss [7].

Some researchers put their focus on novel building blocks for high performance neural network architectures. He et al. introduced residual connection into convolutional network [8]. Zhang et al. proposed Resunet for semantic segmentation neural network [9]. Alom et al. proposed recurrent residual convolutional neural network with U-Net as base architecture and recurrent residual convolutional unit as building block [10]. Szegedy et al. proposed inception and residual inception block to build the network architecture [11]. Oktay introduced attention gate into U-Net. With attention gate, models implicitly learned to suppress irrelevant regions while highlighted salient features of the target [12].

Handcraft features have also been introduced into U-Net to build network architecture. Zhou et al. proposed to build oriented response network block using Active Rotating Filters (ARFs) [13]. Liu et al proposed to build multi-level wavelet CNN using Harr wavelets based on U-Net architecture, with discrete wavelet transformation in up-sampling path and inverse discrete wavelet transformation in down-sampling path [14]. Dai et al proposed deformable convolution and deformable pooling to augment the spatial sampling locations in the convolutional modules [15]. Oyallon et al proposed to construct hybrid architecture using scattering networks [16]. Luan et al proposed Gabor convolutional network which incorporated Gabor filters into CNNs to enhance the resistance of deep learned features to the orientation and scale changes [17]. Tan et al proposed 3D-GLCM CNN which generated multiple 3D-GLCM feature images first and train them using multi-channel CNN model [18].

There are some variations having been used in medical image segmentation. Fabian et al introduced dropout layers in encoder pathways and three segmentation layers in decoder pathways of 3D U-Net for kidney segmentation [19, 20]. Yu et al employed residual block and soft dice together with cross entropy for prostate segmentation [21]. Sabarinathan proposed hyper vision net with attention unit on baseline 2D U-Net to segment kidney [22]. Haghighi et al proposed two-step U-Net for kidney segmentation with first step detecting kidney location and second step doing segmentation [23]. Couteaux et al proposed to segment kidney cortex using 2-dimensional U-Net [24]. Zhao et al proposed to use 3D U-Net with multiple-level supervisions at decoder pathways to segment kidney [25]. All the methods reported good performances.

In this paper, we proposed a novel ULBNet to reduce the semantic gap and improve performance on kidney segmentation from multiple phase CT images. The proposed architecture includes new skip connections of local binary convolution (LBC) [26]. We also proposed a novel strategy for fast retraining a model so that it can adapt well to new coming datasets without manually dataset labelling as prerequisite to start a new task.

II. Method

The network architecture proposed in this paper was shown in Fig. 2. For U-Net, there was a skip connection from output of each layer in encoder pathways to input of each layer in decoder pathways. In our architecture, a local binary convolution (LBC) [26] layer was added to the skip connections. Via those connections, the texture features of encoder layer were expected to keep and pass to the decoder layer of the same level for segmentation.

Local binary pattern (LBP) [27] was simple but powerful texture feature extractor and had wide applications in the area of computer vision. For each pixel, a bit map was first extracted on its $n \times n$ neighborhood via binarization referring to the center pixel (x_L, y_L). Then, LBP at the pixel was calculated by converting the bit map to a decimal number with base two:

$$LBP(x_L, y_L) = \sum_{i=0}^{L-1} 2^i t(x_i - x_L)$$

$$where \quad t(x_i - x_L) = \begin{cases} 1 & x_i \geq x_L \\ 0 & x_i < x_L \end{cases} \quad and \quad L = n^2 - 1 \qquad (1)$$

Xu et al designed local binary convolution (LBC) layer to implement LBP [26]. It started from a binary convolution layer with fixed anchor weights randomly generated with equal probability as 1 or -1, in Bernoulli distribution based on a pre-set sparsity. Sparsity was the percentage of non-zero elements. Following the binary convolution layer was a non-linear activation layer and a 1×1 convolution layer. It was calculated:

$$y_l^c = \sum_{i=1}^{m} w_{l,i}^c \cdot \sigma_{relu}(\sum_s a_i^s * x_{l,i}) \qquad (2)$$

where *a* was the anchor weights, *s* was number of binary filters, *m* was input channels and *c* is output channels. In this paper, we adopted Xu's implementation.

Redesign skip connections

The proposed skip connection performed the computation of

$$x_{decoder} = \sigma_{relu}\,(x_{encoder} + LBC(x_{encoder}, w_{encoder})\,) \qquad (3)$$

where $x_{encoder}$ was feature map in encoder pathway, $x_{decoder}$ was output of skip connections at decoder side, and $w_{encoder}$ was the

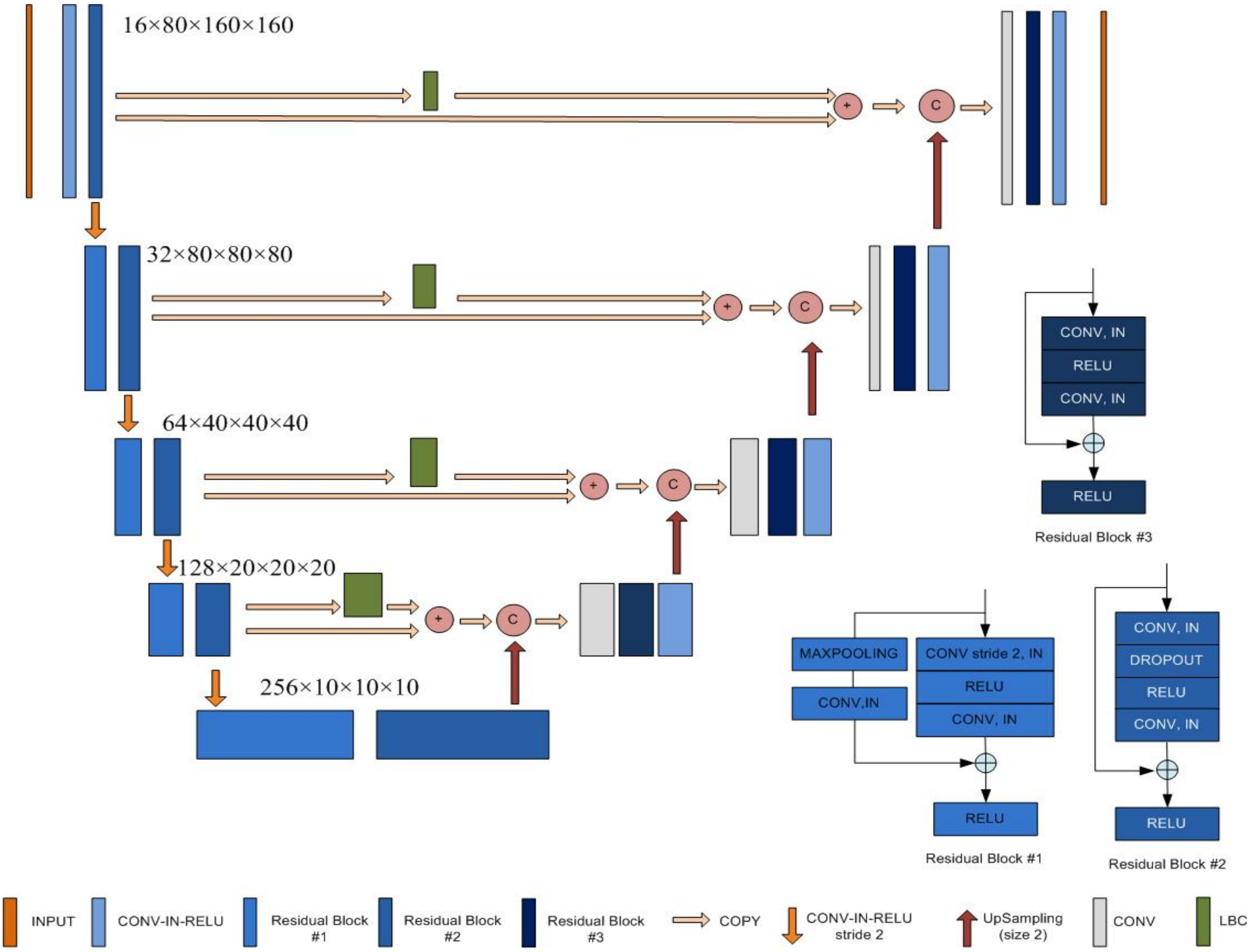


Fig.2. Network Architecture

weights of LBC layers. It changed the connection between the encoder and decoder pathways. Features at encoder pathway underwent an extra LBC block to reach decoder pathways, which brought the handcraft LBP texture feature of the encoder pathways to the correspondence in the decoder pathways. Thus, receptive field was increased and the network generalization was potentially improved.

Network Architecture

The proposed network was a 5-layers U-Nets encoder-decoder architecture with residual block to substitute for the traditional 3-dimensional convolution layer. Two types of residual block were used. Those used in encoder pathways were implemented as Conv3D »» Instance Normalization »» ReLU »» Conv3D »» Instance Normalization »» Dropout »» ReLU. Those used in decoder pathways, the dropout layer was removed. They were implemented as Conv3D »» Instance Normalization »» ReLU »» Conv3D »» Instance Normalization »» ReLU. The dropout rate was set 0.2. Up sampling was done using Upsampling3D and down sampling was done with stride convolutions. The patch size was same as images size and batch size was 1, thus, Instance normalization was used. At first, the network started with 16 feature maps, with same resolution as input. The filter number was doubled with each down sampling operation in the encoder pathways and halved with each up sampling operation in the decoder pathways. As the input shape was not a cube, a stride of (1, 2, 2) was employed on first layer. Except for the first layer, the feature map in each layer was always down sampled by a factor of 2. Similarly, the feature map was always up sampled by a factor of 2 in the decoder pathways. A stride of (1, 2, 2) was employed to up sample the feature map to the last decoder layer. The activation we used was sigmoid. The optimizer we used was Adam. The loss function was dice coefficient:

$$L(Y,\hat{Y}) = -\frac{2|Y \cap \hat{Y}|}{|Y|+|\hat{Y}|} \qquad (4)$$

where $\hat{Y}$ was prediction and Y was ground truth.

Retraining Strategy

For a new task, data and labeling were generally limitations in the applications of deep CNN. Manually labelling was very tedious and time consuming. In this paper, we proposed a strategy to retrain a model for the new task without requiring manually labelling of the new coming datasets.

In this paper, (1) we initially trained a model using public available dataset close to our target; (2) the model was used to predict the datasets of the specific task; (3) we selected a certain portion of the best predictions and used those image/prediction pairs as training data to retrain the model. After several iterations of step (3), that is, retraining model by repeat substituting for best prediction results in the training set, the predictions were able to visually close to the desirable.

The rationale behind was that the expert knowledge on good segmentation had been accumulated in the predictions via human selection for iterations. Initially, the prediction from pre-trained model may not be as same as manually labelled ground truth. However, those human selected good segmentation indicated that expertise knowledge on what to segment on the new datasets had be included, even though the knowledge may be incomplete initially. Such knowledge provided, to a certain extent, the guidance on segmenting the new coming datasets during model retraining. Thus, the prediction can be improved. For datasets, the visually best predictions were assumed consisting relatively complete expertise knowledge compared with those not well segmented. They were used to replace those in the training set and to guidance the model retraining at the next iteration. The knowledge was thus accumulated after each usage of the retrained model and finally resulted in desirable predictions.

In this work, we proposed to first train a model, i.e. model_0, for kidney segmentation with public available corticomedullay phase CT datasets and labels, then employ the model on multiple phase CT datasets for segmentation. Here, the model without retraining were not able to result in an acceptable accuracy on new coming datasets which were different from the original training datasets. Model retraining was necessary. For retraining, the top 20% ~ 40% segmented results on multiple phase CT images using visual quality control were selected to form a new training set. The 40% was preferred if good results within predictions were sufficient. A subset of same amount of public CT images were randomly extracted from training set. This subset was also included in the new training set. The size of the new training set was totally 480 ~ 640.

We re-trained the network on the new training datasets started from model_0. The flowchart was illustrated in Fig 3. The validation set remained the same and new coming datasets were used as the test set. The datasets with top predictions were selected to compose the new training set. With the new training set, model_0 evolved to model_1 after one epoch. Model_1 was then used to predict the new coming datasets. If results need further improvement, the dataset with top predictions were selected to compose new training datasets and another model retraining was conducted. Similarly, the top 20% ~ 40% predictions on multiple phase CT images were added in the training sub-set and training for one epoch to obtain model_2. In our study, two iterations were able to result in desirable performance on segmenting kidney in multiple phase CTs.

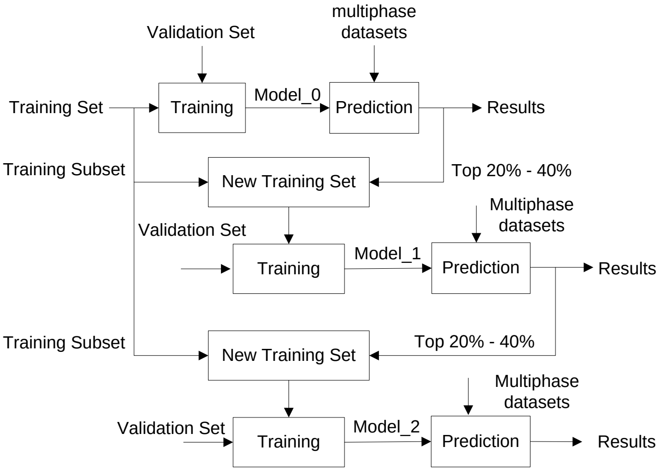


Fig. 3. The flowchart of the proposed retraining strategy

III. EXPERIMENT AND RESULTS

Our work started with training a model on public KiTS19[1] datasets, which consisted of 210 corticomedullay phase CT images. Based on the model obtained, we aimed to a much general model that is able to segment multiple phase CT images acquired in the study for renal stone treatment using the proposed retraining strategy. Our in-house datasets consisted of 35 anonymized multiple phase CT images, acquired using GE OPTIMA CT600 CT scanner with the slice collimation of 0.625 mm, matrix of 512×512 pixels.

A. *Image Preprocessing*

Resolution preprocessing: Generally, voxel spacing of CT images were inhomogeneous. The datasets in CNN were viewed as array and their resolutions were not processed during training. Thus it was required to resample all datasets to the normalized before feeding them into the network. To compromise the limitation of GPU memory and high resolution images requiring much memory, all the 210 datasets were resliced to $3.22mm \times 2.03mm \times 2.03mm$, and cropped to $80 \times 160 \times 160$ empirically.

Data Augmentation: Data augmentation was critical for training discriminative CNNs. We adopted the data augmentation strategy being tested in [28]. The 210 datasets were split into 135 for training, 33 for validation, and 42 for testing. Our image augmentation strategy applied on 135 training datasets. There were no data augmentation strategy employed on the 33 validation datasets and 42 testing datasets. Initially, all testing datasets were corticomedullay phase CTs, after satisfied model obtained, we added multiple phase CTs for

[1] https://kits19.grand-challenge.org/

prediction and evaluation. The 35 in-house datasets were split into 21 for retraining strategy, 7 for validation, and 7 for testing.

Flips: A CT volume in training set was flipped in left-right, anterior-posterior, and superior-inferior direction respectively. Even though the position of the patients were following certain standard during CT scanning and seldom position flip occurred, we still believe that the organs were recognizable after flip and their relationship with other organs persisted. Such transformations help model learn the target in different aspects and finally benefit for recognizing target in real situations.

Jittering: For each CT volume in training set, a 1 ~ 4 density value in Hounsfield was added on or subtracted from each voxel.

Scaling: Each CT volume in training set was scaled from 90% to 110%. We increased/decreased the entire size simultaneously. We did not scale the volume in individual dimensions as in [7].

Gaussian Blur: Each CT volume was filtered by a Gaussian function with mean of 0 and variance from 0.1 to 0.9.

Rotations: Each CT volume was rotated clockwise or counter clockwise 10 ~ 20 degrees in axial plane.

Shears: Each CT volume in training set was sheared in the range of [0.1, 0.35].

The data augmentation was implemented using simple ITK[2]. Each augmentation was done by varying its transformation parameters across the range specified above. In the end, the 135 training datasets were expanded to a final augmented training set of 5400 CT volumes. The parameter ranges were determined empirically and the guidelines were that the transformations resulted from them should be able to make CT volumes still explainable. In the 21 multiple phase images for retraining strategy, datasets were augmented only when they were selected to go into new training set. The rest were not augmented.

Image Normalization: All the training set, validation set and testing set were cast to window level (-120, 300) where -120 was standard Hounsfield value of fat and 300 was of contrast-enhanced CT. The voxel with density lower than -120 was set to -120 and with density higher than 300 was set to 300. Then the whole volume was normalized by its mean density and standard deviation. Different from the corticomedullay phase CT images, the plain phase and excretory phase CT images were processed using a window level of (-120, 200). Similarly, mean and standard deviation of 32.5 were used to normalize the density. The nephrographic phase images were processed same as the corticomedullay phase images. It was important to use different window level on CT images acquiring in different phases, thus, their normalized density distributions were closed as much as possible to guarantee there being a certain accuracy in initial prediction.

B. *Training Procedure*

The patch size was set to 80×160×160 and the batch size was set to 1 due to the limitation of GPU memory. The learning rate was initialized as 3×10^{-4}, and drops by a factor of 0.2 if the loss was not improved in 30 epochs. The training stopped early if training loss did not improve in 50 epochs. Training was done on Nvidia Quadro RTX 5000 (single GPU training). All network architectures were implemented on the tensorflow framework.

C. *Results on public corticomedullay phase image datasets*

In our initial training on public corticomedullay phase datasets, 5-fold cross validation was used to evaluate the model's performance, and dice coefficients were used to measure the accuracy. The individual test accuracy resulted from ULBNet was plotted in Fig. 4. It can be observed that the modal's performance was relative stable and the lowest accuracy was above 95.5%.

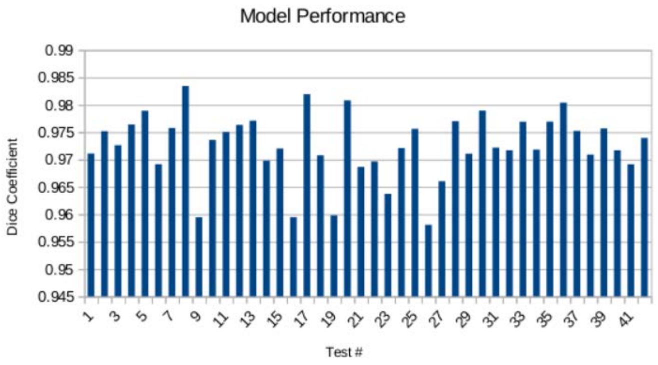


Fig 4. Testing accuracy of ULBNet on the corticomedullay phase image datasets.

With 16 basic filters, ULBNet resulted in an average accuracy of 97.3% on kidney segmentation in the corticomedullay phase CT images, while Resunet resulted in an accuracy of 97.2% as shown in Table I. The proposed ULBNet performed a slightly better than Reunet.

TABLE I. PERFRORMANCE ON PUBLIC CORTICOMEDULLAY PHASE CT IMAGES

model #	Accuracy	
	With data augmentation	Without data augmentation
ULBNet^{16}	0.9729	0.9644
Resunet^{16}	0.9723	0.9637
ULBNet$^{12.1}$	0.9715	0.9634
ULBNet^{12}	0.9714	0.9633
Resunet^{12}	0.9712	0.9627

ULBNet^{16} is ULBNet with 16 basic filters. ULBNet$^{12.1}$ is ULBNet with 12 basic filters and LBC layer in residue block, ULBNet^{12} is ULBNet with 12 basic filters. Resunet^{16} and Resunet^{12} are Resunet with 16 and 12 basic fitlers respectively.

As we used LBP handcraft features as extra skip connections, we naturally considered to use them in the residual block. We added one LBC layer to the residual block and the new block calculated as:

[2] http://www.simpleitk.org/

$$y_l = \sigma_{relu}(x_l + residule(x_l, w_l) + LBC(x_l, w_l)) \quad (5)$$

where x_l, y_l, w_l were input, output and weights of the block. It was tested on the architecture with 12 basic filters due to the GPU memory limit. The results showed that its performance was better than both ULBNet and Resunet with 12 basic filters, but lower than both ULBNet and Resunet with 16 basic filters. Thus, we adopted ULBNet with 16 basic filters for the following experiments in this paper.

For medical image analysis, people argued that the common data augmentation, such as flip, and skew, may not comply with the real situation in the clinical study. It was interesting to train the model with data augmentation and without data augmentation and to evaluate its performance on clinic datasets. We conducted the experiments. The results in Table I showed that model trained with data augmentation had better performance than without data augmentation. It can be explained that data augmentation do help the network to learn to differentiate the target in the viewpoint of human perception by introducing transformed images. As datasets after data augmentation are not real clinical datasets, it is important that data augmentation only employed in training process, while neither of the validation datasets nor the test datasets should include images processed by data augmentation in all experiments in this paper.

D. Results on multiple phase CT images

Based on the model trained using public corticomedullay phase images, retraining strategy was employed to generalize the model to kidney segmentation on multiple phase CT images. The model was tested on multiple phase images and the results were shown in Table II. An accuracy of 97.97% was achieved, and the accuracy on public corticomedullay phase images was 97.17%.

TABLE II. MODEL PERFRORMANCE ON MULTI-PHASES DATASETS

Model	Accuracy	
	multiple phase images	public corticomedullay phase images
Before retraining	0.9349	0.9729
After retaining	0.9797	0.9717

It was interesting to know how well ULBNet and Resunet adapted to multiple phase images. We compared their overall performances on segmenting kidney from multiple phase CT images, as well as individual phase images, as illustrated in Table III. Before retaining strategy applied, it can be observed that performance of ULBNet superseded to Resunet on both segmenting multiple phase and single phase images. The ULBNet, by nature, has good property of generalization for kidney segmentation compared with Resunet. After retraining strategy applied, both ULBNet and Resunet models learned to segment kidney from multiple phase CT images, and ULBNet still performed better than Resunet. Such generalization property of ULBNet will be quite useful in applications where model need to handle new coming datasets with dynamic distributions, and where the data size of the new coming datasets were limited.

TABLE III. MODEL COMPARISION ON GENERALIZATION

Accuracy	Before retraining		After retraining	
	ULBNet^{16}	Resunet^{16}	ULBNet^{16}	Resunet^{16}
Overall	0.9349	0.8854	0.9797	0.9749
PP	0.8927	0.8033	0.9813	0.9767
CMP	0.9711	0.9685	0.9784	0.9720
NP	0.9710	0.9692	0.9764	0.9744
EP	0.9367	0.8864	0.9809	0.9739

In addition, ULBNet performed well on in-house corticomedullay and nephrographic phases even without retraining strategy employed. It indicated that ULBNet can be generalized well to multiple center datasets of similar distributions. There were big space for improvement on plain phase and excretory phase CT images. In the following, some example results of plain phase and excretory phase were plotted in Fig. 5-7 for illustration of model improvement via retraining strategy. The model, trained on public corticomedullay phase images, was marked as model_0 in the paper.

We applied the trained modal_0 on multiple phase CT images for predication. One visually best result in plain phase was shown in Fig. 5. It can be observed that the prediction cover much regions in kidney. Even though there were still under segmentation at certain small area and over segmentation on the boundary, we believe the presented information on kidney segmentation was useful for network to learn kidney appearance in plain phase. A test plain phase CT images was used to illustrate how the kidney prediction had been improved during the retraining process. In Fig. 6, model_0 resulted in true positives near the boundary region, while false negative rate was high. When model_1 learned kidney appearance during plain phase to a certain extent, the true positives started to grow toward inside while false negative regions shrink. Model_2 learn even more kidney appearance compared to model 1 and the true positives resulting from it covered almost the entire kidney region. From visually observing segmentation evolution in Fig.6, the model's performance on segmenting kidney got obvious improvement on plain phase CT images after employing retraining strategy.

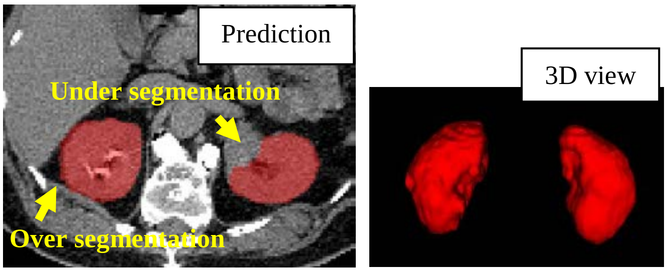


Fig 5. Visually good segmentation on plain phase images using model_0.

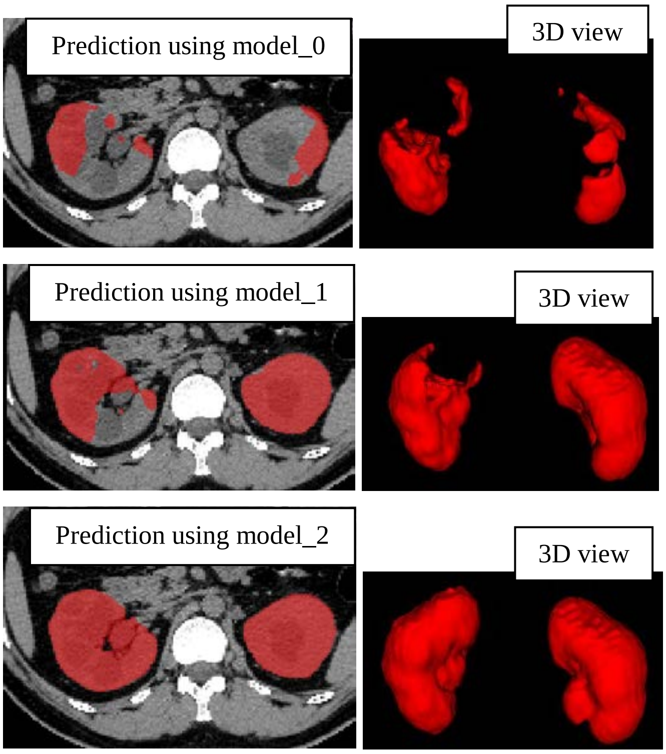


Fig.6. Kidney segmentation on plain phase CTs using model_0, model_1 and model_2 respectively.

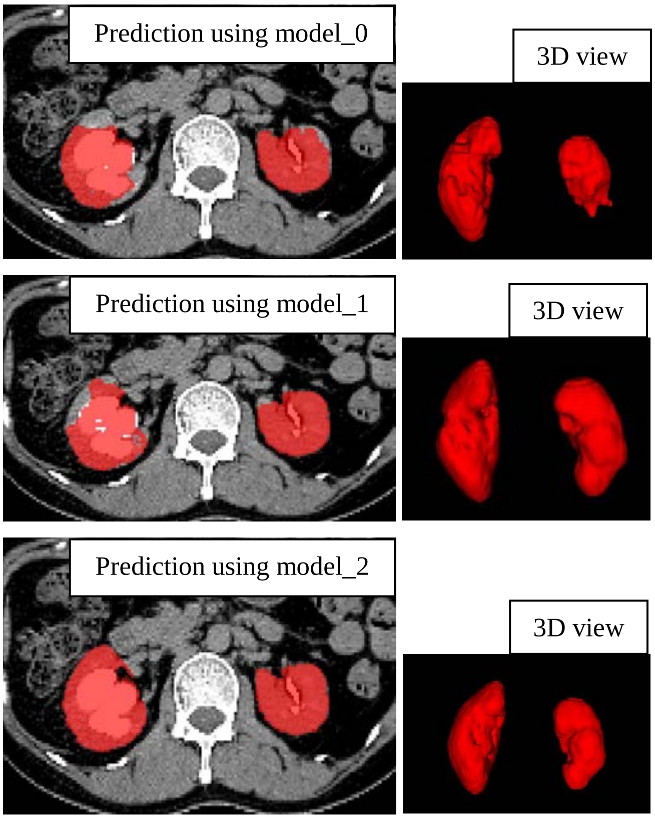


Fig.7. Kidney segmentation on excretory phase CT images using model_0, model_1 and model_2 respectively.

A similar performance evolution can be observed on CT images in excretory phase as shown in Fig. 7, where the true positives grew from inside kidney to the boundary.

IV. Discussions and Conclusions

In this paper, we proposed to add a new skip connection on Resnet to bring the handcraft LBP texture features from encoder pathways to decoder pathways at same level. It made the network generalize well to the multiple phase images. The reason can be that multiple phase kidney images may have common inherit texture features of the kidney in different phases. Such features were captured by the LBC layer and passed to decoder pathways for segmentation.

We proposed a novel retaining strategy that can train a model for a new task without requiring manually data labelling. This strategy was benefit for the deep learning application where data and labelling were limited, especially, at the initial stage of the task and benefit for saving labors. Our retaining strategy required model_0 being able to predict new datasets to a certain extent, say there are at least 20% good predictions can be selected for retraining.

In this paper, limited to the GPU memory, the image size was $80 \times 160 \times 160$, and the batch size was one. To retain high resolution images, we once considered to use training images with one individual kidney while with highest image resolution. However, the results were not as good as expected and only an accuracy of 94.93% was achieved. The possible reason can be that the spatial symmetry of two kidneys were lost when only one kidney were learned, which exerted adverse impacts on the model performance.

Even though images generated using augmentation were not real clinical studies, the data augmentation was adopted in this work because experiments showed it helped to improve model performance. We limited its application to training process only to help network to learn the human perception on the object recognition. It was not used in validation and testing process. That is, the validation datasets and test datasets were all real clinical studies.

Generally, we proposed an ULBNet and a retraining model strategy for kidney segmentation on multiple phase CT images. The method obtained good performance on generalizing to cope with new coming datasets with different density distribution when there was a lack of datasets and labels.

Acknowledgment

The work is partially funded by A*star funding ACCL/19-GAP035-R20H.

References

[1] B. I. Yuh, R. H. Cohan, "Different Phases of Renal Enhancement: Role in Detecting and Characterizing Renal Masses During Helical CT", The American Journal of Roentgenology, 1999

[2] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”. LNCS. 9351. 234-241. 10.1007/978-3-319-24574-4_28, The 18th International Conference on Medical Image Computing and Computer-Assisted Intervention -- MICCAI, 2015

[3] O. Cicek, A. abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D u-net: learning dense volumetric segmentation from sparse annotation. MICCAI, 2016

[4] Z. Zhou, M. Siddiquee, N. Tajbakhsh, J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation”, IEEE Trans. Medical Imaging, pp. 1856-1867, 2020

[5] V. Iglovikov, A. Shvets, “TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation”, arXiv:1801.05746, 2018

[6] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Kohler, T. Norajitra, S. Wirkert, K. H. Maier-Hein, “nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation”, arXiv: 1809.10486, 2018

[7] F. Milletari, N. Navab, S. A. Ahmadi, “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation”, 3DV, 2016

[8] K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition”. IEEE Computer Vision and Pattern Recognition (CVPR), 2016

[9] Z. Zhang, Q. Liu, Y. Wang, “Road Extraction by Deep Residual U-Net”, IEEE Geoscience and Remote Sensing Letters, 2017

[10] Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, V. K. Asari, “Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation”, arXiv:1802.06955, 2018

[11] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017

[12] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, D. Rueckert, “Attention U-Net: Learning Where to Look for the Pancreas”, MIDL, 2018

[13] Y. Zhou, Q. Ye, Q. Qiu, J. Jiao, “Oriented Response Networks”, CVPR, 2017

[14] P. Liu, H. Zhang, W. Lian, W. Zuo, “Multi-level Wavelet Convolutional Neural Networks”, IEEE Access, 2019

[15] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, “Deformable Convolutional Networks”, ICCV, 2017

[16] E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Julien, M. Blaschko, E. Belliovsky, “Scattering Networks for Hybrid Representation Learning”, TPAMI, 2018

[17] S. Luan, B. Zhang, S. Zhou, “Gabor Convolutional Networks”, TIP, 2018

[18] J. Tan, Y. Gao, Z. Liang, W. Cao, M. Pomeroy, Y. Huo, L. Li, M. A. Barish, A. F. Abbasi, P. Pickhardt, “3D-GLCM CNN: A 3-Dimensional Gray-Level Co-Occurrence Matrix-Based CNN Model for Polyp Classification via CT colonography, TMI, 2020

[19] I. Fabian, M-H. Klaus, “An attempt at beating the 3D U-Net,” ArXiv abs/1908.02182 , 2019

[20] I. Fabian, P. Kickingereder, W. Wick, M. Bendszus, M-H, Klaus, “Brain Tumor Segmentation and Radionics Survival Prediction: Contribution to the BRATS 2017 Challenge”. The International Conference on Medical Image Computing and Computer-Assisted Intervention -- MICCAI Brain Lesion Workshop, 2017

[21] L. Yu, X. Yang, H. Chen, Q., Jing, P-A, Heng, “Volumetric ConvNets with Mixed Residual Connections for Automated Prostate Segmentation from 3D MR Images,” Thirty-First AAAI Conference on Artificial Intelligence, 2017

[22] D. Sabarinathan, M. Parisa Beham, S.M.Md.Mansoor Roomi, “Hyper Vision Net: Kidney Tumor Segmentation Using Coordinate Convolutional Layer and Attention Unit”, ArXiv abs/1908.03339, 2019

[23] M. Haghighi, S. K. Warfield, S. Kurugol, “Automatic Renal Segmentation in DCE-MRI using Convolutional Neural Networks”, IEEE 15th International Symposium on Biomedical Imaging (ISBI), 2018

[24] V. Couteaux, S. Si-Mohamed, R. Renard-Penna, O. Nempont, T. lefevre, A. Popoff, G. Pizaine, N. Villain, I. Bloch, J. Behr, M. –F. Bellin, C.Roy, O. Rouviere, S. Montagne, N. Lassau, L.Boussel, “Kidney Cortex Segmentation in 2D CT with U-Nets Ensemble Aggregation”, Diagnostic and Interventional Imaging, pp.211-217, 2019

[25] W. Zhao, Z. Zeng, “Multi Scale Supervised 3D U-Net for Kidney and Tumor Segmentation”. The International Conference on Medical Image Computing and Computer-Assisted Intervention -- MICCAI KiTS19 Grand Challenge, 2019

[26] F. J. Xu, V. N. Boddeti, M. Savvides, “Local Binary Convolutional Neural Networks,” IEEE Computer Vision and Pattern Recognition (CVPR), 2017

[27] M. Pietikainen, A. Hadid, G. Zhao, T. Ahomen, “Computer Vision Using Local Binary Patterns”, Springer, 2011

[28] Z. Hussain, F. Gimenez, D. Yi, D. Rubin, “Differential Data Augmentation Techniques for Medical Imaging Classification Tasks”. Annual Symposium proceedings. AMIA Symposium, pp. 979–984, 2017