Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation

Course Project for CS 736

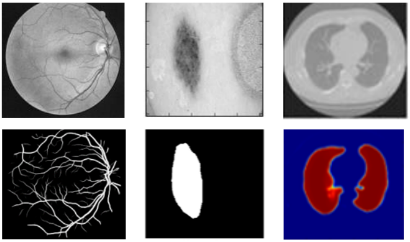
Akkapaka Saikiran   Rohan Shah   Rushabh Kanadiya

Indian Institute of Technology, Bombay

June 26,2020

The problem: Biomedical Image Segmentation

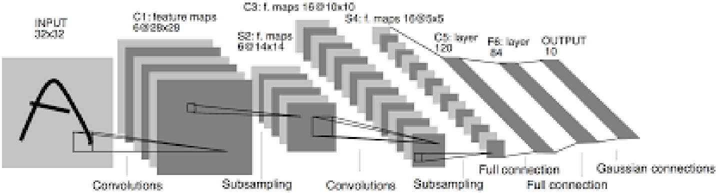
- Segmentation is the process of partitioning an image into different meaningful segments. In medical imaging, these segments often correspond to different tissue classes, organs, tumors, or other biologically relevant structures.
- In the previous decade or so, deep learning techniques have received vast attention for the task of automated image segmentation.



Deep Learning methods in Segmentation

CNNs

Convolutional Neural Networks are a class of deep learning algorithms most commonly applied to analysing image data. They have been around since a long time[1], but saw a resurgence lately[2] and now are the state of the art algorithms for image classification, object detection and image segmentation.



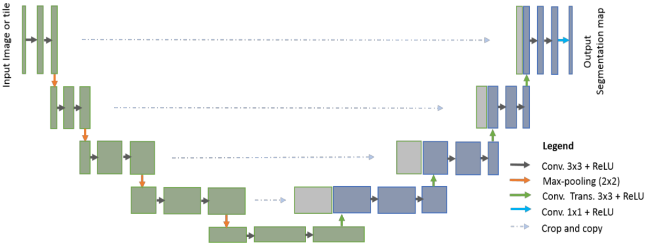
[1] LeNet (1998)

[2] AlexNet (2012), GoogLeNet (2014), ResNet (2015)

Deep Learning methods in Segmentation

U-Net

The U-Net, developed in 2015, is a "fully convolutional" CNN. The main idea is to supplement a usual contracting network by successive layers, where pooling operations are replaced by upsampling (by using fractionally strided convolutions) operators, with a final convolutional which layer can then learn to assemble a precise output.



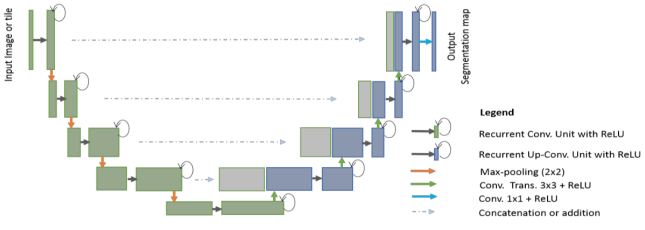
A standard U-Net

R2U-Net: Recurrent Residual U-Net

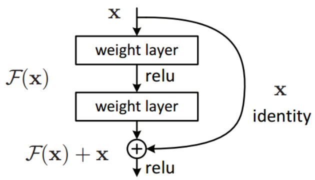
This architecture, the contribution of the paper we base our work on, is a modification to the U-Net. It employs two key ideas.

- Recurrent connections: The key idea of any recurrent block is to reuse weights or maps and maintain some state. In our case, we feed the output of a convolution layer back into that layer as an input a few times (parameterizing the process) before passing it on to the next layer
- Residual connections: These are skip connections, which are used to allow gradients to flow through a network directly, without passing through non-linear activation functions. If $\mathcal{F}$ is a non-linear map (corresponding to a couple of convolutions in our implementation), then the final output $\mathcal{H}$ of a residual block with input $\mathbf{x}$ would be $\mathcal{H}(\mathbf{x}) = \mathcal{I}(\mathbf{x}) + \mathcal{F}(\mathbf{x})$, where I is the Identity function, save for adjusting for shape.

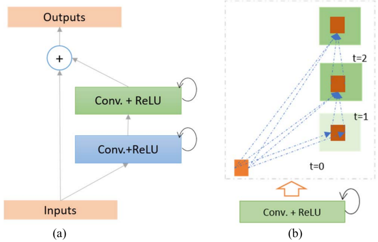
Relevant computational blocks of R2U-Net



A RU-Net, with convolutions replaced by their recurrent equivalents



The residual block from the original ResNet paper



Recurrent Residual block

Implementation Details

- The main convolutions are all 3×3 convolutions with padding and stride 1, and the max pooling which follows it has pool size 2.
- We up-sample using 3×3 transpose convolutions with padding and stride 2, so that the activation maps double in size.
- We double the filter size of the convolutions after each max pool operation, and keep it same within a convolutional block. Filter sizes: $16 \to 32 \to 64 \to 128 \to 256 \to 128 \to 64 \to 32 \to 16 \to 2$.
- We use ReLU activation throughout for non-linearity after each convolution. After every convolution we apply batch normalisation before the non-linear activation function. Dropout is performed in the network after each block.
- We coded various metrics for evaluation. These are: Accuracy, F1-score, Sensitivity, Specificity, Dice Coefficient, and Jaccard Similarity.

Differences and advantages compared to U-Net

Differences

- Residual connections are employed in the main convolutional blocks.
- There is feature accumulation with respect to different time-steps due to the recurrent convolutional layers.
- This model doesn't crop-and-copy, it uses just concatentions while making the skip connections.

Advantages

- A residual unit helps accelerate training, increases depth of network as opposed to width, and mitigates the effect of Vanishing Gradient problem.
- Feature accumulation with recurrent convolutional layers ensures better feature representation for segmentation tasks.
- The recurrent and residual operations, despite not increasing the number of network parameters, have a significant impact on training and testing performance.

Experimental setup and results

Skin Cancer dataset

This dataset is taken from the Kaggle competition on skin lesion segmentation that occurred in 2017.
We could not train on this dataset completely because of limited resources. But here some result images from the preliminary testing.

Results of Preliminary Testing
accuracy: 0.9503 - F1: 0.8485 - SE: 0.8341 - SP: 0.9860 - JS: 0.7737 - DC: 0.8485 - val accuracy: 0.8930 - val F1: 0.6348 - val SE: 0.5782 - val SP: 0.9956 - val JS: 0.5609 - val DC: 0.6349

Experimental setup and results

Blood vessel segmentation

We combined data from three sources (STARE, DRIVE, and CHASE) to train on 68 images in a patch-based fashion, by resizing all images to 256×256 and extracting 11500 128×128 patches, inspired by the original paper, to solve the class imbalance problem.

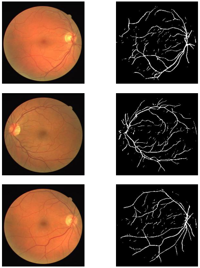


Table 1: Results on a few test images. The first column shows input images, and the second column shows results by R2U-Net

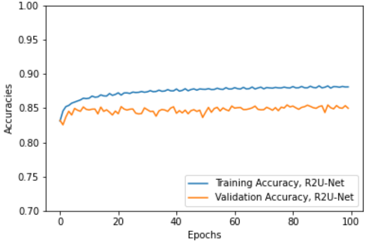


Table 2: Accuracies for R2U-Net

Experimental setup and results

Blood vessel segmentation

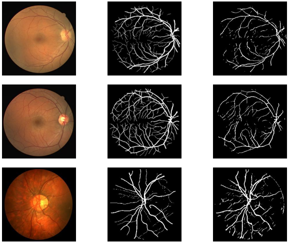


Table 3: Results on a few validation images. The first column shows input images, and the second column shows ground truth, third column shows results by R2U-Net

Metric	Train	Validation
Accuracy	0.882333	0.854549
F1	0.631616	0.489198
JS	0.541661	0.385061
DC	0.631805	0.489393
SE	0.578190	0.485667
SP	0.989355	0.990516

Table 4: Peak values of several metrics

Conclusion

- We implemented two algorithms to segment medical images, one the standard U-Net and another by adding recurrent residual connections to it. R2U-Net performed better than U-Net with similar number of parameters.
- Data Augmentation was performed on ISIC dataset and was found to be imperative to the good results.
- Weighted Cross Entropy Loss helps tackle class imbalance problem and speeds up training.
- Moreover, Dropout with p=0.5 was incorporated which showed definite improvements, as observed from the retinal blood vessel segmentation.
- Accuracy was discovered to be a misleading metric because of the class imbalance, other metrics(Jaccard Similarity, Dice Coefficient) are better indicators of performance.
- For Implementation, Pure Tensorflow functions were found to be much faster because of better integration and using parallel computing, caching etc.

Conclusions

- One interesting idea we considered but did not code yet was incorporating bottlenecks in the network, inspired by Google's Inception modules, using 1×1 convolutions to decrease number of parameters of the network, and increase non-linearity to mitigate over-fitting and get better performance.
- Image wise standardization enhances the image contrast, and it helped us in retinal blood essel segmentation, normalization over the dataset did not work out to be helpful.