

# Preliminary Study on Deep Iterative Semantic Segmentation

Diana Teixeira e Silva<sup>1,2</sup>  
up201805131@edu.fe.up.pt  
Ricardo Cruz<sup>1,2</sup>  
rpcruz@fe.up.pt  
Tiago Gonçalves<sup>1,2</sup>  
tiago.f.goncalves@inesctec.pt

<sup>1</sup> Faculdade de Engenharia  
Universidade do Porto  
Porto, Portugal  
<sup>2</sup> INESC TEC  
Porto, Portugal

## Abstract

Semantic segmentation consists of classifying each pixel according to a set of classes. This process is particularly slow for high-resolution images, which are present in many applications ranging from biomedicine to the automotive industry. In this work, we propose an algorithm targeted to segment high-resolution images based on two stages. During stage 1, a lower-resolution interpolation of the image is the input of a first neural network, whose low-resolution output is resized to the original resolution. Then, in stage 2, the probabilities resulting from stage 1 are divided into contiguous patches, with the less confident ones being collected and refined by a second neural network. The main novelty of this algorithm is the aggregation of the low-resolution result from stage 1 with the high-resolution patches from stage 2. We use the U-Net as the segmentation model and evaluated our proposal in three databases. Our method shows similar results to the baseline regarding the Dice coefficient, with fewer floating-point operations per second.

## 1 Introduction

Segmenting images using neural networks can be time-consuming and require a lot of memory, especially when working with high-resolution images, common in the biomedical area due to high-resolution digital microscopes and in autonomous driving applications.

Neural networks for segmentation, such as the U-Net [6], produce a probability, for each pixel, of belonging to the region of interest. A common practice, when the image is high resolution, is to split the image into patches and process each patch separately [8]. Such approach has two problems: (1) it spends as much time in hard-to-segment regions as it does in easy-to-segment regions where there is nothing of relevance; (2) the boundary between patches is a problem and has to be dealt with in a special way.

Therefore, the idea is to produce an iterative segmentation method for neural networks, whose simplified overview is presented in Figure 1. Iterative segmentation methods already exist [1, 2, 4, 9], but their focus is on improving the quality of the segmentation, not the speed. Our proposal is applicable to every type of high-resolution image, but it is tested over two types of images (biomedical and autonomous driving).

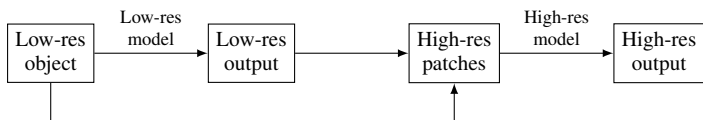


Figure 1: Simplified overview of the work.

## 2 Related Work

Image segmentation is the process of clustering an image into regions of interest (ROI) so that every pixel belonging to an ROI should be similar in terms of several characteristics (e.g. color, texture, shape or intensity) [3, 7]. The development of novel artificial intelligence techniques allows us to divide image segmentation methods into traditional and deep-based approaches. Traditional segmentation techniques are *iterative*, often rely on domain knowledge and benefit from feature engineering techniques to achieve their final results. Examples of these approaches are: thresholding, edge-based, region-based, deformable models or graph cuts.

On the other hand, when it comes to neural networks, segmentation is trained in an iterative manner and inferred in one step. A popular model

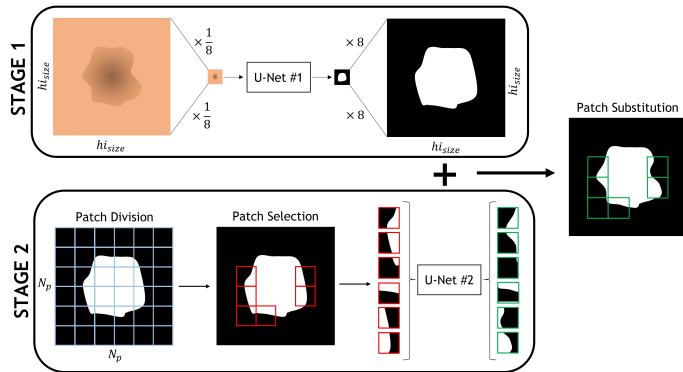


Figure 2: Two-stage segmentation.

is the U-Net, which relies on the skip-layers that directly connect the feature maps of the encoder to the analogous feature maps of the decoder, preventing the loss of information [6].

## 3 Proposal

The proposal consists of elaborating a new segmentation method based on two stages: (1) using a U-Net to segment a low-resolution version of the image; (2) based on the probabilities produced by this U-Net, identify poorly segmented image patches and use a second U-Net to refine these patches.

The proposed idea is illustrated in Figure 2, where an image is down-sampled by a relevant factor and segmented by the first U-Net. The resulting probability map is then upsampled to the original resolution and divided into patches. The patches from areas where the activation map is less confident are collected and each one is refined by the second U-Net. Finally, the outcome of this algorithm is the aggregation of the low-resolution result from stage 1 with the high-resolution patches from stage 2.


## 4 Experiments

**Data:** The datasets used contain images from dermoscopy (PH2), fundus imaging (RETINA) and autonomous driving (KITTI). Further details regarding each dataset, such as the number of images (N), the average image resolution (Avg Res) and the average percentage of foreground values relative to the entire image (%Fg) are presented in Table 1.

Every dataset was randomly divided into 70% of the total images being training data and 30% being test data. To homogenize the resolution, all the images from the same dataset were resized to a square image with the length of  $h_{i\_size}$  – an even value, preferably a power of 2, close to both image dimensions (width and length) of the average resolution of the dataset. The  $h_{i\_size}$  value used for each dataset is also presented in Table 1.

**Model:** The encoder path of both U-Nets architectures contained five 2D convolutions, with kernels of size  $3 \times 3$  and stride 2, followed by the activation function ReLU. The number of kernels started at 64 and doubled at each layer. The decoder path consisted of five 2D transposed convolutions with equal kernel size and stride from the contracting path, each preceding a ReLU function. The number of kernels was reduced in half for each layer of the expansive path. A  $1 \times 1$  convolution operation was applied in the final layer to obtain 1 as the output number of classes.

Table 1: Datasets for semantic segmentation.

Dataset	N	Avg Res	$hi_{size}$	%Fg	Example
PH2*	200	575×766	768	31.4	
RETINA†	66	745×782	768	7.3	
KITTI‡	200	375×1271	512	6.6	

\* <https://www.fc.up.pt/addi/ph2%20database.html>

† Composition of three datasets:

<https://blogs.kingston.ac.uk/retinal/chasedb1>

<https://cecas.clemson.edu/~ahoover/stare>

<https://drive.grand-challenge.org>

‡ <https://www.cvlibs.net/datasets/kitti>

**Training:** The following data augmentation transformations were applied: horizontal flip; contrast/brightness modification between -0.1 and 0.1; and random rotation, with limits -180° and 180°. This last transformation was only applied to the biomedical datasets, which are dermoscopic and retinal images and therefore the model should be invariant to rotation. Every aforementioned transformation had a probability value of 0.5.

The loss used consisted of an unweighted sum between two losses,  $\mathcal{L}(y, \hat{y}) = \mathcal{L}_f(y, \hat{y}) + (1 - D(y, \hat{y}))$ , where  $\mathcal{L}_f$  is the focal loss [5] and the other term corresponds to the inverse of the Dice coefficient. The focal loss is a weighted version of cross-entropy that is helpful in such situations of imbalance (notice in Table 1 that the %Fg values are well below 50%). In fact, some of our tests showed it worked better than vanilla cross-entropy. The focal loss parameters  $\gamma$  and  $\alpha$  used were 2 and 0.25, respectively, corresponding to the recommended values from the authors. The inverse Dice coefficient is not as smooth as a loss function, but we also minimized it since it is the metric that we used.

Adam was used as the optimizer with a batch size of 64, and a learning rate of 0.0001, trained for 200 and 1000 epochs in the case of stage 1 and stage 2, respectively. The model of stage 2 was trained for more epochs since in each epoch only a small region of each image was selected.

## 5 Results

For each dataset, the average Dice coefficient and FLOPS were computed to evaluate the model’s performance, where the baseline (BL) model represents the traditional approach consisting of the same U-Net architecture using the original images as input.

Figure 3 shows the Dice results for the different experiments. As expected, there is usually a noticeable gain between stage 1 (S1) and stage 2 (S2). Increasing the number of patches  $N_p$  either results in a plateau or reduced performance.

Figure 4 depicts the FLOPS performed by the neural networks. When increasing the number of patches, the number of operations reduces because, while stage 1 is the same, stage 2 operates with smaller patches (that is, fewer convolutions are necessary, even if more patches are being used). In all cases, our method requires fewer operations than the baseline. However, in practice, the overhead introduced by extracting the patches and merging them with the initial segmentation might produce results that are not as optimistic.

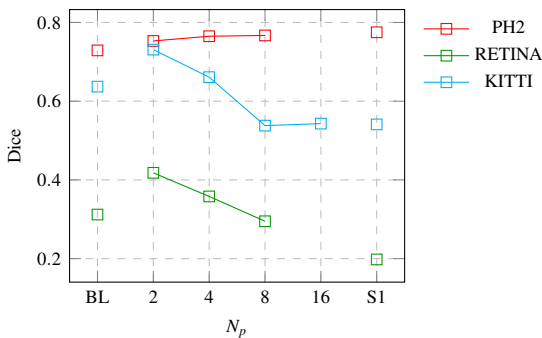


Figure 3: Average Dice coefficient.

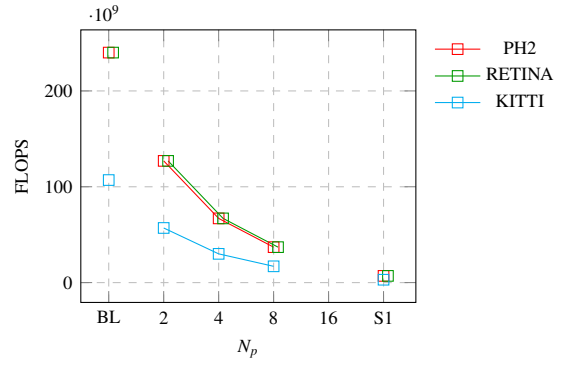


Figure 4: Number of floating-point operations per second.

## 6 Conclusion

We validated the proposed strategy in 2 biomedical datasets and 1 dataset regarding autonomous driving. Our algorithm provided overall Dice coefficients similar to the baseline, not showing noticeable gains. When calculating the number of arithmetic operations performed by the neural networks, it becomes clearer that the method is able to have a similar performance to the baseline while saving on the number of operations.

## Acknowledgments

This work is supported by European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project n° 047264; Funding Reference: POCI-01-0247-FEDER-047264], and by National Funds through the Portuguese funding agency FCT – Fundação para a Ciência e a Tecnologia – within the PhD grant “2020.06434.BD”.

## References

- [1] Andrea Banino, Jan Balaguer, and Charles Blundell. PonderNet: learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.
- [2] Kelwin Fernandes, Ricardo Cruz, and Jaime S Cardoso. Deep image segmentation by quality inference. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.
- [3] Rafael C Gonzalez and Richard E Woods. Digital image processing, prentice hall. *Upper Saddle River, NJ*, 2008.
- [4] Jung Uk Kim, Hak Gu Kim, and Yong Man Ro. Iterative deep convolutional encoder-decoder network for medical image segmentation. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 685–688. IEEE, 2017.
- [5] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [6] 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*, pages 234–241. Springer, 2015.
- [7] Neeraj Sharma, Lalit M Aggarwal, et al. Automated medical image segmentation techniques. *Journal of medical physics*, 35(1):3, 2010.
- [8] Chang Wang, Zongya Zhao, Qiongqiong Ren, Yongtao Xu, and Yi Yu. Dense U-Net based on patch-based learning for retinal vessel segmentation. *Entropy*, 21(2):168, 2019.
- [9] Wei Wang, Kaicheng Yu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Recurrent U-Net for resource-constrained segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2142–2151, 2019.