

Two-stage Semantic Segmentation in Neural Networks

Diana Teixeira e Silva^{1,2}, Ricardo Cruz^{1,2}, Tiago Gonçalves^{1,2}, Diogo Carneiro³

¹ Faculty of Engineering, University of Porto, Portugal

² INESC TEC, Porto, Portugal

³ Bosch Car Multimedia, Braga, 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 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 propose the U-Net architecture segmentation, evaluated in six databases. Our method shows similar results to the baseline regarding the Dice coefficient, with fewer arithmetic operations.

Keywords: autonomous driving, bioengineering, deep learning, iterative inference, semantic segmentation.

1. INTRODUCTION

Segmenting images using neural networks can be time-consuming and requires a lot of memory, especially when working with high-resolution images [1], 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 [2], 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 [3]. 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 [4]–[7], 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).

Besides this Introduction, the paper is organized as follows: Section 2 covers the related work; Section 3 explains the proposed algorithm; Section 4 presents the details of our implementation; Section 5 provides and discusses the results and Section 6 concludes the paper and suggests future work directions. The code related to this paper is publicly available¹.

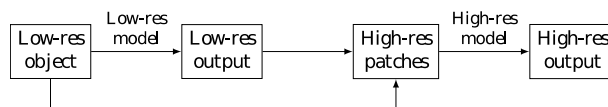


Figure 1. Simplified overview of the work.

¹ <https://github.com/dianartsilva/two-stage-segmentation>

2. RELATED WORK

2.1 Image Segmentation

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) [8], [9]. 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 [10]. Examples of these approaches are: thresholding [11], [12], edge-based [13], [14], region-based [15]–[18], deformable models [19], [20] or graph cuts [21], [22]. Deep-based approaches apply deep neural networks, often trained end-to-end (i.e., these frameworks receive an image and output a segmentation mask). Long *et al.* [23] proposed the Fully Convolutional Network architecture, considered the pioneering successful approach to deep-based image segmentation. Following this methodology, Ciresan *et al.* [24] and Ronneberger *et al.* [2] proposed improved models for biomedical use-cases, being the last (i.e., U-Net) one of the most widely used backbone architectures. The popularity of U-Net relies on the short and long skip-connections, which directly connect the feature maps of the encoder to the analogous feature maps of the decoder, preventing the loss of information [2]. Nevertheless, there are other methodologies that are worth our attention: the Global Convolutional Network [25], proposed by Peng *et al.*, distinguishes from its predecessors by using large convolutional kernels; and DeepLabv3+, proposed by Chen *et al.* [26], revisited the concepts of *trous convolution* and *spatial pyramid pooling*. In all these models, segmentation is trained in an iterative manner and inferred in one step.

2.2 Iterative Approaches

Applying deep-based models to high-resolution images is computationally expensive. For instance, in computational pathology, one of the main limitations is related to the large file size due to the high-resolution (and different magnifications) of the *whole slide images* [27], [28]. In these cases, it is common to employ a *multiple instance learning* strategy [29], where the large inputs are divided into smaller inputs [30]. These smaller inputs are processed by a deep-based model and the individual results are aggregated by a scoring function. On the other hand, another possible solution may be to implement iterative methodologies, thus revisiting the traditional image segmentation pipeline. Following this idea, previous works have tried to make the U-Net iterative. Fernandes *et al.* [4] propose having a neural network that acts as an oracle: given an image and segmentation pair, the oracle is trained to predict a score of their fitness. On inference, the oracle is used to iteratively modify the segmentation in the direction that improves the score. Kim *et al.* [5] propose a recurrent version of the same idea: the image and segmentation are received as a pair and the neural network predicts a new segmentation. This step can be repeated as many times as desired so that the new segmentation is again used as input to the model, producing a new segmentation. This idea is improved by Wang *et al.* [6] and generalized to any task by Banino *et al.* [7].

There are works that focus on improving training and/or inference time, typically by working with multiple scales, but not on segmentation tasks. For example, due to computational constraints, Google AI performs alpha matting on mobile devices (i.e., extracting a foreground object) by a two-stage process whereby a neural network performs an initial step, and a secondary network is used only on any areas that might require further work [31]. A similar approach has been used for range estimation by Miangoleh *et al.* [32]. There are works that have used multiple scales for segmentation, but the focus is on the metric performance, not on reducing inference time [1], [33]. One recent work proposes combining iterative clustering and deep learning, but again the focus is on the metric performance [34].

In many bioengineering applications, images are very large (high-resolution) and segmentation is costly. For that reason, patch-based methods are common. In these approaches, images are divided into several, smaller patches that are possible for neural networks to process [3]. Our proposal is motivated by these works, but it is inspired by the techniques mentioned in the previous paragraph. Similar work to our own, but for the purpose of image classification, is [35], where they use a neural network with multiple scale inputs; these inputs are decided by attention mechanisms.

3. PROPOSAL

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

The proposed idea is illustrated in Figure 2. The input image, with an initial resolution of $hi_{size} \times hi_{size}$ (see Table 1), is downsampled by a factor of $1/8$ in each image dimension (width and length). This transformed image is the input of U-Net #1, whose output is a low-resolution probability map which is then upsampled to the original resolution. This high-resolution probability map is divided into non-overlapping contiguous square patches with size hi_{size}/N_p , where N_p is the number of patches in each image dimension (for instance, $N_p = 6$ divides the image into $N_p^2 = 36$ equal-sized patches, 6 in each dimension, as observed in Figure 2). Each patch is evaluated regarding its average pixel-value (*mean*): the N_p patches which reported the lowest $|mean - 0.5|$ values are then selected for U-Net #2. The final segmentation result is given by the output of U-Net #1 with the refined patches obtained by U-Net #2.

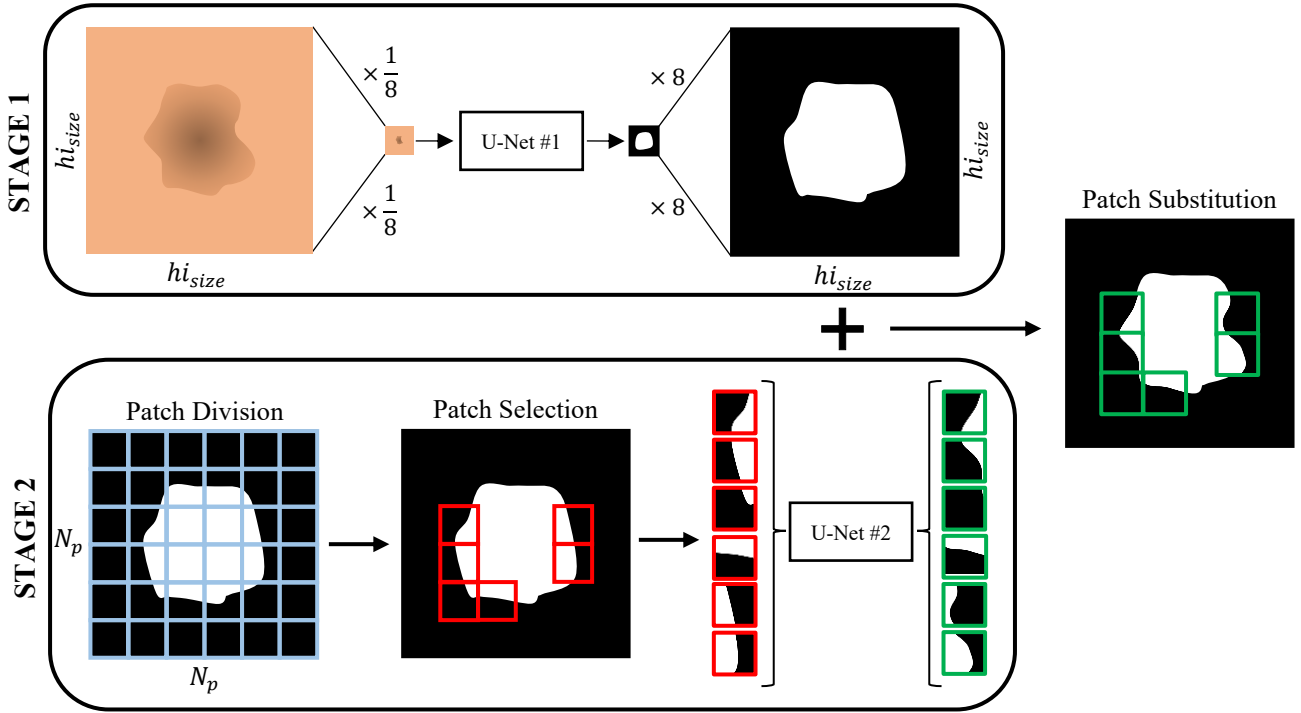


Figure 2. Two-stage segmentation.

4. IMPLEMENTATION

4.1 Data



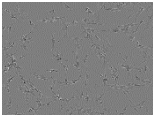
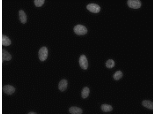


To validate the performance of the developed method, we used data from 6 different datasets – 4 covering biomedical images and 2 covering driving scenarios – in order to ensure the algorithm implemented was data-independent. Further details regarding each dataset, such as the total number of images (N), the average image resolution (Avg Res) and the average percentage of foreground values relative to the entire image (% Fg) are presented in Table 1.

The biomedical datasets contain dermoscopic (PH2), fundus imaging (RETINA), and histological (SARTORIUS and BOWL2018) images. The goal of the SARTORIUS dataset is to segment neuronal cells from phase-contrast microscopy images, while BOWL2018 aims to segment the nuclei from different types of cells. The images of the latter dataset were acquired under a large set of conditions regarding magnification and modality (bright-field/fluorescence).

The autonomous driving datasets used were KITTI and BDD100K, comprising real driving scenarios acquired from different locations. In these datasets the task is to segment vehicles. In all cases, the task is binary segmentation.

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 hi_{size} – an even value, preferably a power of 2, close to both image dimensions of the average resolution of the dataset. The hi_{size} value used for each dataset is also presented in Table 1.

Table 1. Datasets for semantic segmentation.

Dataset	N	Avg Res	hi_{size}	% Fg	Example
PH2 [36]	200	575 x 766	768	31.4	
RETINA* [37]–[39]	66	745 x 782	768	7.3	
SARTORIUS [40]	606	520 x 704	512	10.5	
BOWL2018 [41]	670	328 x 369	256	13.6	
KITTI [42]	200	375 x 1271	512	6.6	
BDD100K [43]	≈ 8k	720 x 1280	768	9.5	

* Composition of three datasets: CHASE_DB1, DRIVE and STARE.

4.2 Model

The encoder path of both U-Nets architectures contained five 2D convolutions, with kernels of size 3×3 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 1×1 convolution operation was applied in the final layer to obtain 1 as the output number of classes.

The pooling operation observed in the original model [2] was replaced by convolution operations with stride 2, which downsamples the size of the feature map to half while duplicating the number of feature channels.

4.2.1 Training

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 [44] and the other term corresponds to the inverse of the Dice coefficient (see Section 4.3). 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 γ and α used were 2 and 0.25, respectively, corresponding to the recommended values from the authors.

Adam was used as the optimizer with a batch size of 64 and a learning rate of 0.0001. The models were trained for enough epochs to converge (200 for U-Net #1 and 1 000 for U-Net #2). U-Net #2 was trained for more epochs since in each epoch only a small region of each image was selected.

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 microscopic, dermoscopic and retinal images, therefore the model should be invariant to rotation. Every aforementioned transformation had a probability value of 0.5.

In addition to these, resizing and random cropping operations were added to train each U-Net accordingly to their task:

- *Baseline (BL)*: All images were resized to the hi_{size} described in Table 1.
- *Training U-Net #1 (low resolution)*: Images were resized to $(1/8 + 5\%)hi_{size}$, followed by random cropping of dimensions corresponding to $1/8$ of hi_{size} .
- *Training U-Net #2 (patches)*: Images were resized to hi_{size} in both dimensions, with a random crop to a square of length $\frac{1}{N_p}hi_{size}$.

4.3 Metrics

In the case of a segmentation task, metrics such as accuracy (globally and each class individually) and the Dice coefficient are the most used to evaluate a model’s result. The Dice coefficient, also known as the Sørensen index, is a measure that derives from the original quotient of similarity: $Dice = 2 \cdot TP / (2 \cdot TP + FP + FN)$, where TP , FP , and FN are true positives, false positives and false negatives, respectively.

For evaluation of speed, metrics such as inference time – the time required for a deep neural network to apply the trained model to new data – and floating-point operations per second (FLOPS) are suitable for the task.

5. RESULTS AND DISCUSSION

For each dataset, the average Dice coefficient and FLOPS were computed to evaluate the model’s performance for the N_p values 2, 4, 8, and 16. Note that, in Figure 3(a) and Figure 3(b), $N_p = 16$ was not computed for some datasets since the resulting patch size was not divisible by 2^5 , 5 being the number of encoder layers of our U-Net model. The baseline (BL) model represents the traditional approach consisting of the same U-Net architecture using the original images as input.

Figure 3(a) shows the Dice coefficients for the different experiments. These results suggest there are apparently three different behaviors, which might indicate in which cases the method makes more sense to use: (1) PH2 and BDD100K; (2) KITTI and RETINA; (3) SARTORIUS and BOWL2018.

PH2 and BDD100K: Interestingly, these datasets showed a Dice coefficient that was actually better during U-Net #1 (S1) relative to the baseline (BL), but then plateaued. In the case of PH2, lowering the resolution seems to help segment the lesions, in particular the ones which are not so homogeneous and might induce the model to segment them in a not-so-accurate way. The plateau registered might be because U-Net #2 has as input patches that only contain a portion of a skin lesion, therefore losing part of the context and compromising its performance. Contrarily to PH2, the vehicles from the BDD100K dataset could fit entirely in a single patch, hence having better chances of a good segmentation. Because most of the images from this dataset have the vehicles in a center plane, as seen in Figure 4, the division of the total image in 16 patches would be the best-case scenario, being able to segment the center in a way $N_p = 2$ could not, while giving each patch the most context for segmentation when compared to $N_p = 8$.

KITTI and RETINA: As expected, there is a drop in the Dice coefficient value between the baseline (BL) and U-Net #1 (S1). This is particularly pronounced for the RETINA dataset due to the small width of the retinal blood vessels – the reduction of the image resolution by a factor of 8 probably compromises the differentiation of such structures. U-Net #2 greatly improved on S1, actually surpassing the baseline, with the best Dice coefficient for $N_p = 2$. KITTI, a similar dataset to BDD100K, was expected to also present similar behavior. However, the average resolution values for the images in each dataset might play a bigger role in S1, compromising the results of datasets with smaller resolutions. Figure 5 presents the results of our method’s pipeline for a KITTI image.

BOWL2018 and SARTORIUS: Results for these datasets were worse for our method than the baseline. The problem comes from the U-Net #1 (S1) segmentation, as illustrated by the example in Figure 6. It seems to be the case that in datasets containing images with many dispersed cells, which are relatively small, U-Net #1 results are much worse than the baseline. While U-Net #2 is able to improve upon it, it is not sufficient to overcome this fact.

As expected, there is usually a noticeable gain between the U-Net #1 and U-Net #2 models. Increasing the number of patches N_p either results in a plateau or in reduced performance. In addition, S1 is the most critical stage of our method, being particularly influenced by the downsampling factor and the size of the object to segment. Furthermore, the results from the last stage are mostly affected by the patch division and the loss of context from U-Net #2.

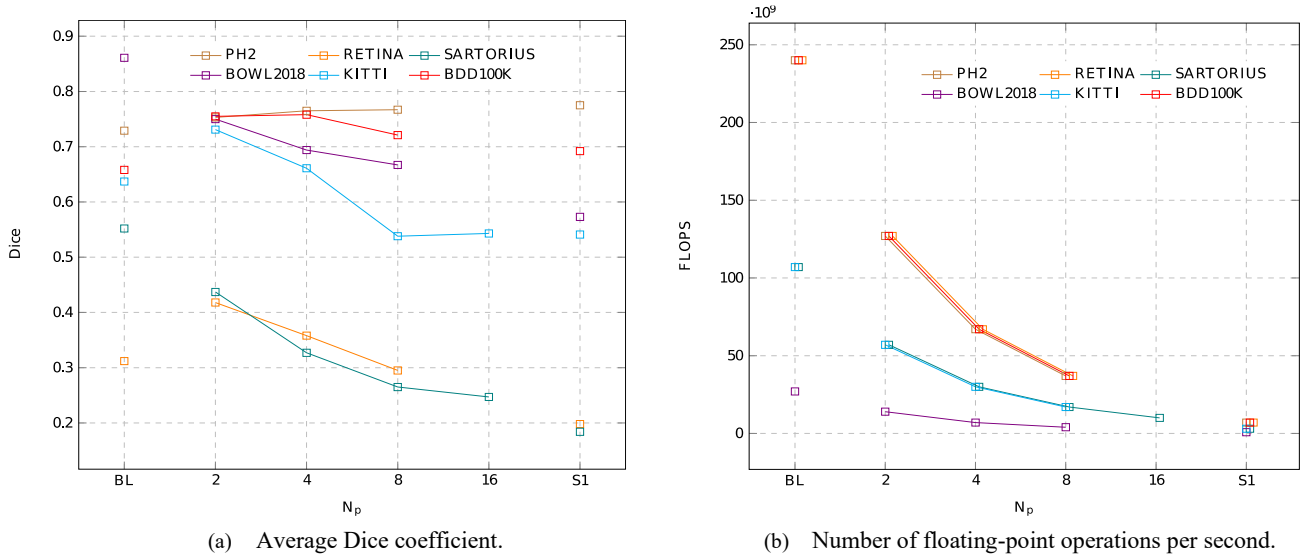


Figure 3. Results (BL is the baseline model, S1 is only the U-Net #1 model).

Figure 3(b) depicts the FLOPS performed by the neural networks, as measured using the profiler from e-Lab². FLOPS per convolution is given by $FLOP_{conv} = F \cdot F \cdot C_{in} \cdot H_{out} \cdot W_{out} \cdot C_{out}$, where $F \cdot F$ is the spatial dimension of the filter, C_{in} and C_{out} are the number of input and output channels, respectively, and $H_{out} \cdot W_{out}$ is the output shape [45]. As the number of patches (N_p) increases, the size of each patch decreases quadratically, $(h_{i_size}/N_p)^2$. Since the architecture is the same, N_p only affects $H_{out} \cdot W_{out}$ along the network. Therefore, when increasing the number of patches, the number of operations reduces since, while both U-Nets are the same, U-Net #2 operates with smaller patches (i.e., 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 result in times that are not as optimistic.

6. CONCLUSION

This work addresses the problem of semantic segmentation of high-resolution images, commonly dealt with in bioengineering and autonomous driving applications. We proposed a two-stage segmentation algorithm aiming to make the segmentation process aforementioned less costly: in stage 1, a low-resolution version of the original image is the input of a first U-Net; the output of this neural network – a low-resolution probability map – is then resized to the original resolution and divided into a relevant number of patches, being the less confident ones identified and refined by the second U-Net in stage 2.

² <https://github.com/e-lab/pytorch-toolbox>

We validated the proposed strategy in 4 biomedical datasets and 2 datasets regarding autonomous driving. Our algorithm provided overall Dice coefficients similar to the baseline (with an average difference between BL and our method of -0.06), not showing noticeable gains. When calculating the number of FLOPS performed by the neural networks, it becomes clearer that the method is able to have a similar performance to the baseline while saving on at least 50% and up to 80% of the number of operations.

For future work, the choice of patches could be improved using a more fine-grained sliding window instead of a contiguous sliding window. One possibility would be to identify regions where the neural network is undecided and use these regions with the same scale – this could be especially beneficial for autonomous driving datasets since the same objects have different sizes (some cars are further away than others) and this would help homogenize the scale of these objects.

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”.

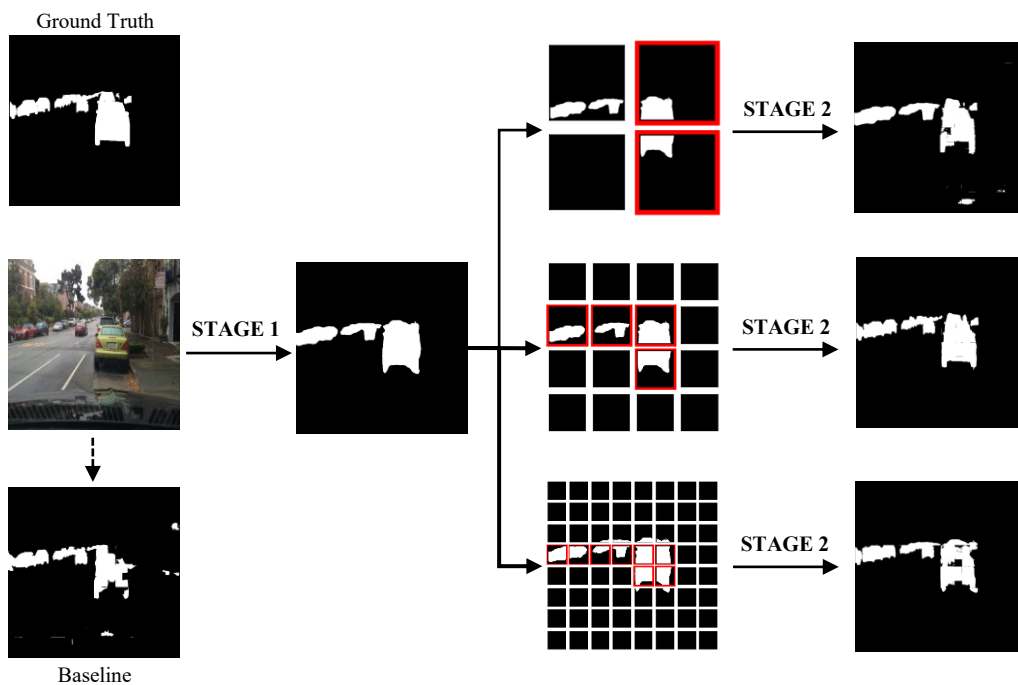


Figure 4. BDD100K example.

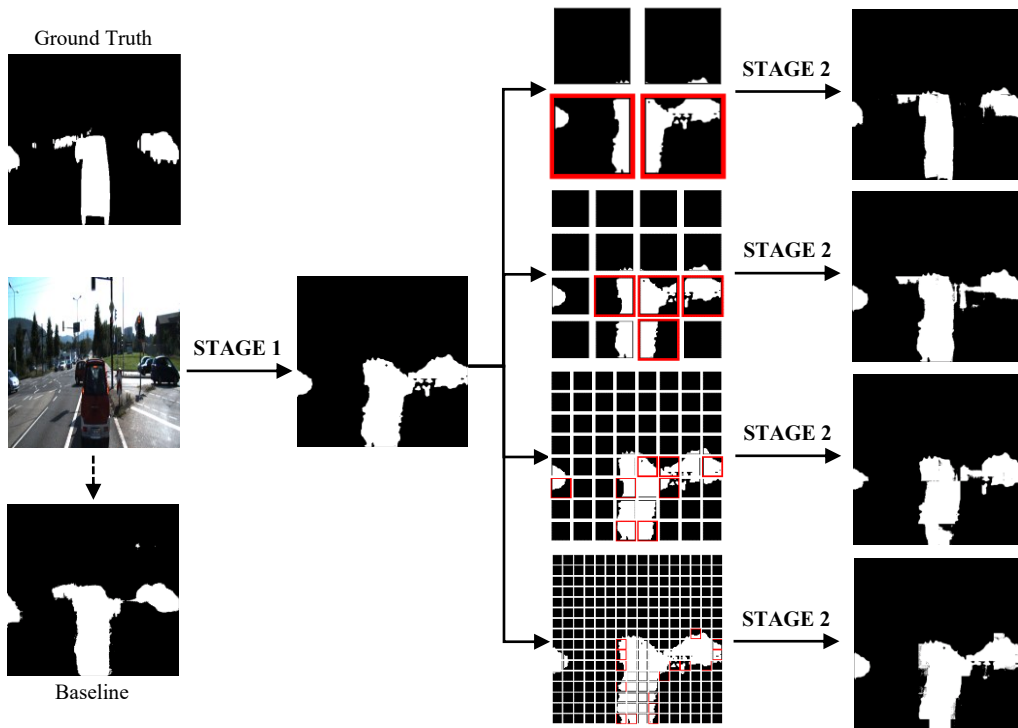


Figure 5. KITTI example.

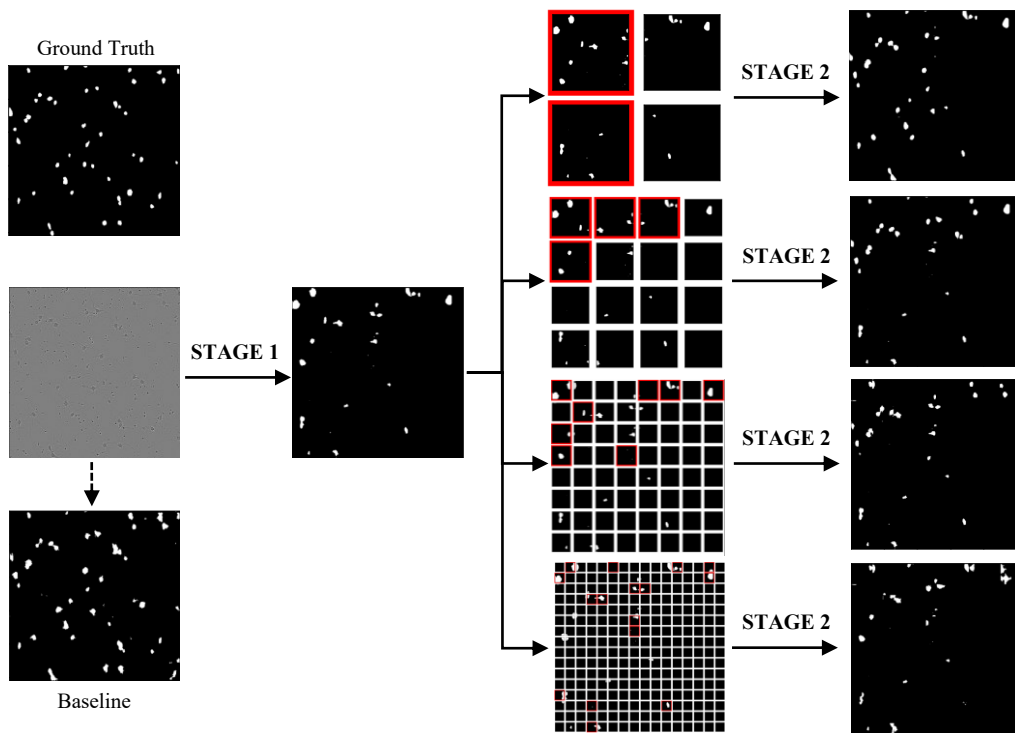


Figure 6. SARTORIUS example.

REFERENCES

- [1] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for Real-Time Semantic Segmentation on High-Resolution Images," Sep. 2018.
- [2] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.
- [3] C. Wang, Z. Zhao, Q. Ren, Y. Xu, and Y. Yu, "Dense U-Net based on patch-based learning for retinal vessel segmentation," *Entropy*, vol. 21, no. 2, p. 168, 2019.
- [4] K. Fernandes, R. Cruz, and J. S. Cardoso, "Deep image segmentation by quality inference," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.
- [5] J. U. Kim, H. G. Kim, and Y. M. 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)*, 2017, pp. 685–688.
- [6] W. Wang, K. Yu, J. Hugonot, P. Fua, and M. Salzmann, "Recurrent U-Net for resource-constrained segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2142–2151.
- [7] A. Banino, J. Balaguer, and C. Blundell, "PonderNet: Learning to ponder," *arXiv preprint arXiv:2107.05407*, 2021.
- [8] R. C. Gonzalez and R. E. Woods, "Digital image processing, prentice hall," *Upper Saddle River, NJ*, 2008.
- [9] N. Sharma, L. M. Aggarwal, and others, "Automated medical image segmentation techniques," *J Med Phys*, vol. 35, no. 1, p. 3, 2010.
- [10] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognit*, vol. 26, no. 9, pp. 1277–1294, 1993.
- [11] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans Syst Man Cybern*, vol. 9, no. 1, pp. 62–66, 1979.
- [12] P. K. Saha and J. K. Udupa, "Optimum image thresholding via class uncertainty and region homogeneity," *IEEE Trans Pattern Anal Mach Intell*, vol. 23, no. 7, pp. 689–706, 2001.
- [13] H. B. Kekre and S. M. Gharge, "Image segmentation using extended edge operator for mammographic images," *International journal on computer science and Engineering*, vol. 2, no. 4, pp. 1086–1091, 2010.
- [14] E. A. Zanaty, "Improved region growing method for magnetic resonance images (MRIs) segmentation," *American Journal of Remote Sensing*, vol. 1, no. 2, pp. 53–60, 2013.
- [15] J. Shan, H. D. Cheng, and Y. Wang, "A completely automatic segmentation method for breast ultrasound images using region growing," in *11th Joint International Conference on Information Sciences*, 2008, pp. 332–337.
- [16] P. R. Tamilselvi and P. Thangaraj, "Segmentation of calculi from ultrasound kidney images by region indicator with contour segmentation method," *global journal of computer science and technology*, 2012.
- [17] E. Day *et al.*, "A region growing method for tumor volume segmentation on PET images for rectal and anal cancer patients," *Med Phys*, vol. 36, no. 10, pp. 4349–4358, 2009.
- [18] J. B. Davis, B. Reiner, M. Huser, C. Burger, G. Székely, and I. F. Ciernik, "Assessment of 18F PET signals for automatic target volume definition in radiotherapy treatment planning," *Radiotherapy and Oncology*, vol. 80, no. 1, pp. 43–50, 2006.
- [19] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int J Comput Vis*, vol. 1, no. 4, pp. 321–331, 1988.
- [20] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on image processing*, vol. 7, no. 3, pp. 359–369, 1998.
- [21] V. Thongnuch and B. Uyyanonvara, "Automatic optic disk detection from low contrast retinal images of ROP infant using GVF snake," *Suranaree J Sci Technol*, vol. 14, no. 3, pp. 223–234, 2007.
- [22] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient ND image segmentation," *Int J Comput Vis*, vol. 70, no. 2, pp. 109–131, 2006.
- [23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [24] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," *Adv Neural Inf Process Syst*, vol. 25, 2012.

- [25] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.
- [26] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.
- [27] S. Robertson, H. Azizpour, K. Smith, and J. Hartman, "Digital image analysis in breast pathology—from image processing techniques to artificial intelligence," *Translational Research*, vol. 194, pp. 19–35, 2018.
- [28] M. Cui and D. Y. Zhang, "Artificial intelligence and computational pathology," *Laboratory Investigation*, vol. 101, no. 4, pp. 412–422, 2021.
- [29] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: A survey of problem characteristics and applications," *Pattern Recognit*, vol. 77, pp. 329–353, 2018.
- [30] S. P. Oliveira *et al.*, "Weakly-Supervised Classification of HER2 Expression in Breast Cancer Haematoxylin and Eosin Stained Slides," *Applied Sciences*, vol. 10, no. 14, 2020, doi: 10.3390/app10144728.
- [31] Google AI Blog, "Accurate Alpha Matting for Portrait Mode Selfies on Pixel 6." 2022.
- [32] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy, "Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9685–9694.
- [33] T. Shen *et al.*, "High Quality Segmentation for Ultra High-resolution Images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1310–1319.
- [34] Q. Yu *et al.*, "CMT-DeepLab: Clustering Mask Transformers for Panoptic Segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2560–2570.
- [35] K. Thandiackal *et al.*, "Differentiable Zooming for Multiple Instance Learning on Whole-Slide Images," *arXiv preprint arXiv:2204.12454*, 2022.
- [36] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. S. Marcal, and J. Rozeira, "PH2—A dermoscopic image database for research and benchmarking," in *2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, 2013, pp. 5437–5440.
- [37] J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Trans Med Imaging*, vol. 23, no. 4, pp. 501–509, 2004.
- [38] A. D. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Trans Med Imaging*, vol. 19, no. 3, pp. 203–210, 2000.
- [39] M. M. Fraz *et al.*, "An ensemble classification-based approach applied to retinal blood vessel segmentation," *IEEE Trans Biomed Eng*, vol. 59, no. 9, pp. 2538–2548, 2012, doi: 10.1109/TBME.2012.2205687.
- [40] Kaggle, "Sartorius – Cell Instance Segmentation." 2021.
- [41] Kaggle, "2018 Data Science Bowl." 2018.
- [42] H. Alhaija, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes," *International Journal of Computer Vision (IJCV)*, 2018.
- [43] F. Yu *et al.*, "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," Sep. 2018, [Online]. Available: <http://arxiv.org/abs/1805.04687>
- [44] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [45] S. H. S. Basha, M. Farazuddin, V. Pulabaigari, S. R. Dubey, and S. Mukherjee, "Deep Model Compression based on the Training History," Jan. 2021, doi: 10.48550/arxiv.2102.00160.