Retinal vessels segmentation using Active Learning for data annotation

Federico Germinario
federico.germinario@eurecom.fr

Andrea Ghiglione
andrea.ghiglione@eurecom.fr

Gabriele Gioetto
gabriele.gioetto@eurecom.fr

1. Motivation

In the paper [1] it is proposed a new architecture for classifying 2D patch images in order to know where a vessel is located inside a brain's image and for extracting vessels combining human-annotations, a segmentation network and a classification network. We worked on a similar scenario, but instead of brain images we used eyes images from the DRIVE dataset. In both scenarios there is the problem that labeling vessels at a patch level is time expensive and very often human's expertise is required. For this reason, our work is focused on reducing the burden of the annotation process through Active Learning [6, 2], while maintaining a good performance. In addition to vessel's patches classification, we worked on vessel's segmentation which was a challenging task in our scenario.

2. Method

As we mentioned in section 1 we are working on eyes images. In particular, we work on a patch basis cutting the black borders from the images; *i.e.* we divide our images into small areas and then we use these images as input to the network. A schema of this process is reported in Figure 1. Since Active Learning techniques require an *oracle*, we simulate his actions by saving each patch image specifying either if it contains vessels or not through a set of patch-level annotation maps (this set won't be present in a real scenario, but the knowledge of an oracle will be used). We decided to discard patches which are black (*i.e.* patches that do not contain any parts of the eye and so which are not useful). We also normalized all the patches.

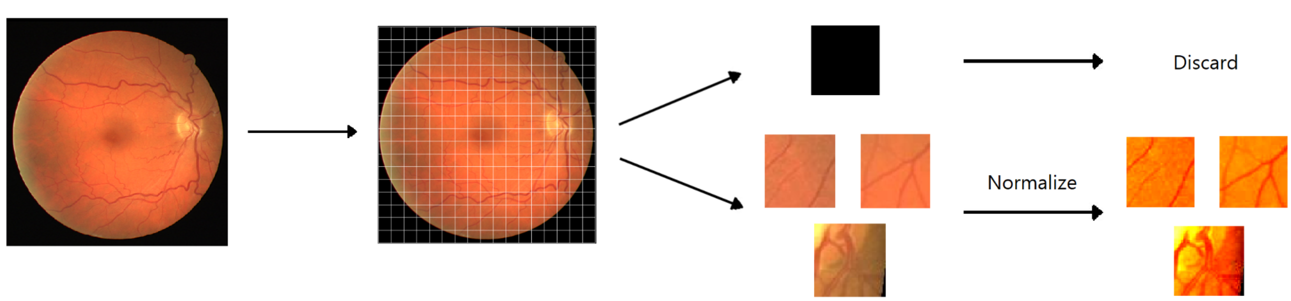

Figure 1. Generating patches from eye's image, discarding full black patches and normalize other patches

In order to lighten the labeling process while maintaining a good performance, we used an Active Learning technique on the classifier that consists in selecting the most useful samples from the unlabeled dataset and send them to an oracle for the annotation (here we simulated the oracle labeling the images). The Active Learning framework for the classifier is represented in Figure [2]. As a classification network, we mainly used the PNET [5] architecture, but we also tried well-known networks as ResNet50 [4] and VGG16 [8], both pre-trained on ImageNet dataset and fine-tuned on eye's images.

As uncertainty measures we implemented Least Confidence and Entropy, that both returns the sample on which the classifier is more uncertain on. The former (1) select the samples that have a classification confidence the most diverse from either 1 or 0. The latter (2) select the samples on which there is an high uncertainty, since entropy is a measure of unpredictability of a random variable. As long as some transformations in the images do not affect the correctness of the presence of vessels in it, some data augmentation is applied to the images. In this way also the active learning framework benefits from having more images generated from one single labeling process by the oracle.

$$argmin_{x_i}|\hat{y}_i - 0.5| \tag{1}$$

$$-\sum_x p(x)log(p(x)) \tag{2}$$

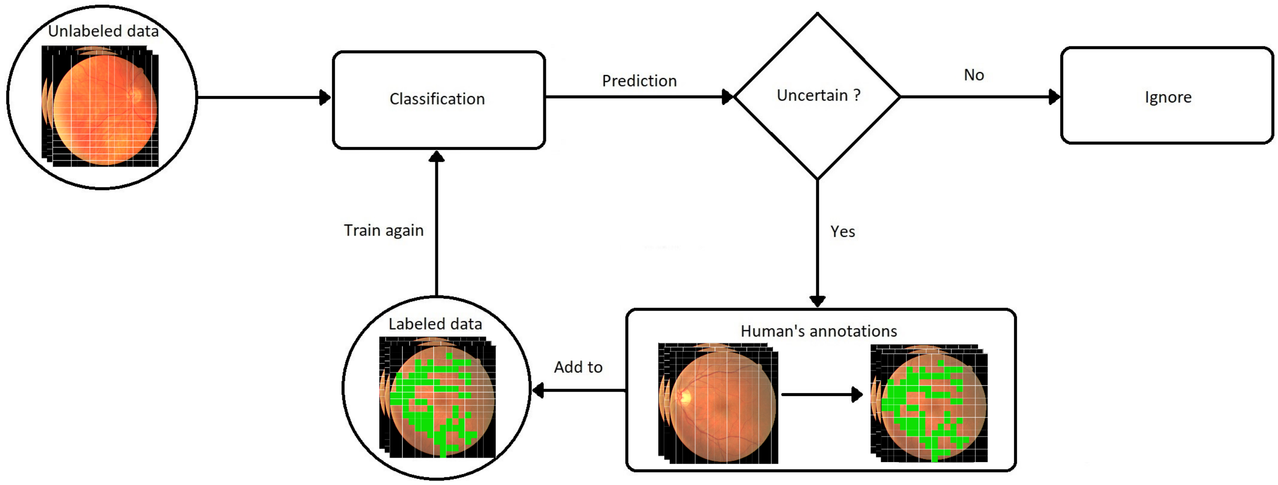

Figure 2. Training a model with Active Learning. Since labeling may be expensive, the algorithm will query the user choosing the best samples to learn from.

Once we generated the patch-level labels our work continued in obtaining a pixel-level classification, *i.e.* we moved to an additional segmentation task. This task is particularly heavy since we are in a weak-supervised scenario. In order to obtain a first approximation of pixel-level labels we used K-means clustering algorithm and OpenCV Canny method. This allowed us to get labels to train a segmentation network for the final result. As segmentation network we exploited the Unet [7] ; in particular we used two 2D-Unets in cascade [3] which was extremely useful since the first Unet will recover rough-mask labels coming from Canny and K-means and the second Unet will produce better segmentations thanks to the output of the first Unet.

3. Experiments

Our dataset was composed by 20 images in total. We decided to do an 80/20 train-test split, therefore there were 16 images in the training set and 4 images in the test set. Doing some exploration on the dataset we noticed that, using patches of size 32x32 pixels, the problem is unbalanced. In particular, the 88% of the patches contain vessels. In order to deal with this problem we tried several techniques. First, we worked with undersampling, making the training set smaller but balanced. However, as shown in Figure 3 and Figure 4 the loss values are very different and this is probably due to the fact that with random under sampling we have a smaller dataset that may decrease performances on the test set. After that we tried oversampling by copying samples of the smaller class, but again this wasn't effective as we could expect since duplicated samples don't mean additional information. Then we tried to work with patches created on a sliding-windows approach, but this method created a more unbalanced dataset; for these reasons we discarded these methods. The only effective method we worked with was to add the possibility to enlarge the training set with an external dataset (CHASE-DB1) selecting only its non-vessels patches. This showed to be effective in reducing the loss. Results can be found in the Appendix 5.

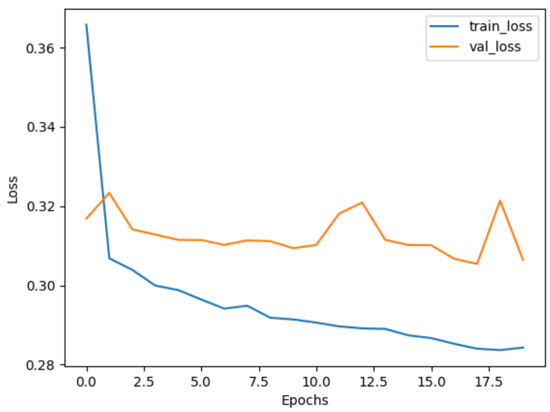

Figure 3. Training with the full dataset, *PNET* architecture

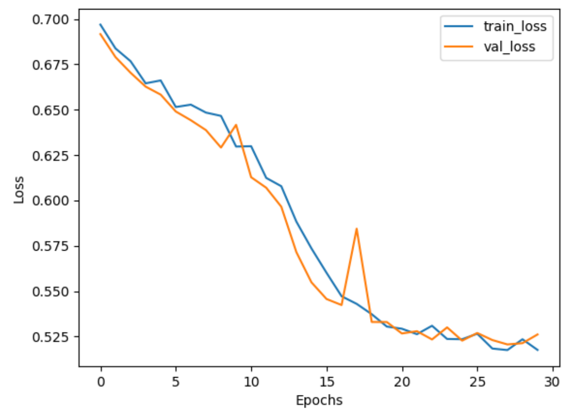

Figure 4. Training with a balanced dataset, *PNET* architecture

Then, we tested how many iterations in an Active Learning framework were needed in order to obtain similar results with respect to the model trained on the full training set. Adding the most uncertain 50 samples per iteration and starting with a training set which was the 15% of the whole dataset, we took 10 iterations to reach good performances on both classes (*i.e.* non-vessels and vessels). This means that it is enough to focus on the most meaningful portion of the dataset in order to achieve the same performance achievable by considering the whole dataset. In particular, in our task active learning was effective on the smaller class (non-vessels); we can accept a low recall and a good precision since this means that the algorithm will classify less patches as non-vessel, but it will classify them with more certainty, which is more desirable than an higher recall and a lower precision in our specific task. We assessed the model's performances paying attention also to precision and recall, rather than accuracy only. We did that since accuracy is not the best metric when the dataset is highly unbalanced. Overall, active learning reached a good performance, saving time that would have been spent in data annotation. Results are reported in Table 1 and Table 2.

	precision	recall	f1-score	accuracy
non-vessel	0.71	0.41	0.52	
vessel	0.92	0.98	0.95	
total				0.91

Table 1. Results with full training set

	precision	recall	f1-score	accuracy
non-vessel	0.73	0.37	0.49	
vessel	0.92	0.98	0.95	
total				0.91

Table 2. Results at 10th iteration of Active Learning

After assessing Active Learning performances, we proceeded in exploiting patches and their patch-level label in order to generate a segmentation at patch level which could be used as ground truth for the final segmentation network. We tried two different approaches:

- **K-means**: This is a clustering technique to segment the patch into K areas. It is based on the concepts of similarity between a cluster's elements and dissimilarities between clusters. We set different K values based on the patch content; if it was non-vessel we set K to 1 (*i.e.* masking it) if it was vessel we set K to $2 + V$ where V is a specificity value we can add to have more precise segmentation. This was done since K-means has shown to be very likely to capture noise around vessels into the vessel's cluster, so an higher K is useful to separate the noise from the vessel in the cluster segmentation. With K-means we used colored patches since the grayscale made harder the distinction between clusters for the algorithm.

- **Canny**: This is a method from the OpenCV library. We chose to use it since it is able to capture the shapes in an image and so it was extremely useful in extracting vessels from the patches. In this case we used a blur grayscale version of the patches, because in this way we removed some noise and we were able to highlight the vessel shape in a better way making it easier to be captured from the Canny method (vessels are the darker part of an image in grayscale). Since in this case we were not constricted to any K value, we used an heuristic to filter possible random mistakes. In particular, if more than 1/4 of a patch was classified as vessel it was probably a mistake, containing noise only and so we ignored that segmentation. Also, we set some tolerance parameters in order not to be too strict in looking for a vessel shape.

Since K-means relies more on colors, while Canny focused more on shapes which are more robust, we chose the latter for creating the labels for the final segmentation network. An example of the advantage of Canny over K-means is reported in Figure 5. In addition, overall results of these two methods are reported in the Appendix 5.

Figure 5. From left to right: normalized image, image segmented with K-means, image segmented with OpenCV's Canny. As we can see, Canny is able to capture also the shape of vessels which are difficult to notice at a first sight

Finally we trained the segmentation network for 40 epochs; thanks to the two Unets we were able to overcome the problem of not having very accurate pixel-level annotations (due to the weak supervised scenario) achieving good results. In particular we obtained a score of 73% as mean intersection-over-union.

One example of segmentation we obtained is reported in Figures [6, 7].

Figure 6. Original image

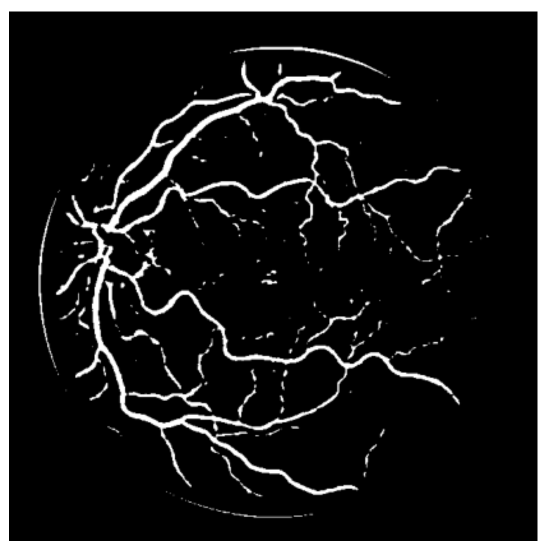

Figure 7. Output of segmentation network

4. Conclusions

Overall, we are satisfied from the results since we were able to:

- **Implement and test the effectiveness of the Active Learning framework**
  Reducing the necessary labels to less than 25% of the data was a noticeable result in both terms of time and resources.

- **Exploit methods to extract vessels in a noisy scenario**
  DRIVE patches are extremely noisy and so noticing and extracting vessels may be extremely hard sometimes. The choice of combining contrasts in grayscale patches, shapes and ad-hoc heuristic was extremely effective with respect to a normal clustering algorithm.

- **Obtain good retina segmentation in a weak supervision scenario**
  Weak supervision is undoubtedly one of the hardest task in semantic segmentation, but we were able to achieve good performances on the bigger vessels. However, further improvement can be done on smaller vessels.

5. Contributions

All the team's members worked on this project; every step that we did was reviewed and approved by the entire team.

Appendix

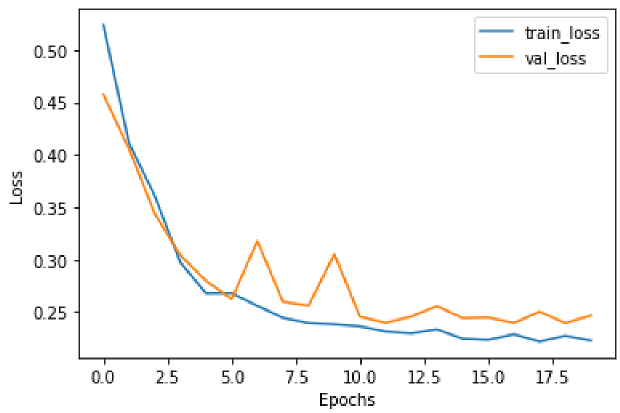

Figure 8. Loss with the addition of a second dataset to enlarge the cardinality of the smaller class in DRIVE dataset

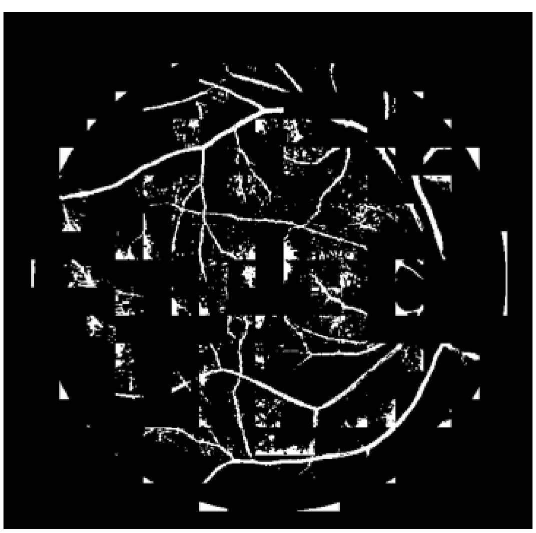

Figure 9. K-means first approximation of pixel-level labels. We can notice how this algorithm captures a lot of noise.

Figure 10. Canny first approximation of pixel-level labels. Here the general shape is captured and it will be improved in the Unet.

References

[1] Vien Ngoc Dang, Francesco Galati, Rosa Cortese, Giuseppe Di Giacomo, Viola Marconetto, Prateek Mathur, Karim Lekadir, Marco Lorenzi, Ferran Prados, and Maria A. Zuluaga. Vessel-captcha: an efficient learning framework for vessel annotation and segmentation, 2021.

[2] Tivadar Danka and Peter Horvath. modal: A modular active learning framework for python, 2018.

[3] Maria Dias, João Monteiro, Jacinto Estima, Joel Silva, and Bruno Martins. Semantic segmentation of high-resolution aerial imagery with w-net models. In Paulo Moura Oliveira, Paulo Novais, and Luís Paulo Reis, editors, *Progress in Artificial Intelligence*, pages 486–498, Cham, 2019. Springer International Publishing.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[5] W A Kampman, J M Kros, T H R De Jong, and M H Lequin. Primitive neuroectodermal tumours (PNETs) located in the spinal canal; the relevance of classification as central or peripheral PNET : case report of a primary spinal PNET occurrence with a critical literature review. *J Neurooncol*, 77(1):65–72, Nov. 2005.

[6] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A survey of deep active learning, 2020.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.