Eurographics Workshop on Visual Computing for Biology and Medicine (2018)
A. Puig Puig, T. Schultz, and A. Vilanova (Editors)

Uncertainty-Guided Semi-Automated Editing of CNN-based Retinal Layer Segmentations in Optical Coherence Tomography

S. Gorgi Zadeh[1], M. W. M. Wintergerst[2] and T. Schultz[3,1]

[1] Department of Computer Science, University of Bonn, Germany
[2] Department of Ophthalmology, University of Bonn, Germany
[3] Bonn-Aachen International Center for Information Technology (B-IT), University of Bonn, Germany

Abstract

Convolutional neural networks (CNNs) have enabled dramatic improvements in the accuracy of automated medical image segmentation. Despite this, in many cases, results are still not reliable enough to be trusted "blindly". Consequently, a human rater is responsible to check correctness of the final result and needs to be able to correct any segmentation errors that he or she might notice. For a particular use case, segmentation of the retinal pigment epithelium and Bruch's membrane from Optical Coherence Tomography, we develop a system that makes this process more efficient by guiding the rater to segmentations that are most likely to require attention from a human expert, and by developing semi-automated tools for segmentation correction that exploit intermediate representations from the CNN. We demonstrate that our automated ranking of segmentation uncertainty correlates well with a manual assessment of segmentation quality, and with distance to a ground truth segmentation. We also show that, when used together, uncertainty guidance and our semi-automated editing tools decrease the time required for segmentation correction by more than a factor of three.

1. Introduction

Deep convolutional neural networks (CNNs) have greatly increased the accuracy of fully automated image segmentation [GLGL18]. Despite this, there are still important reasons to design effective methods for editing the results of automated analysis, especially in a medical context. First, results are usually still not perfect, in particular in the presence of additional pathological changes that may not have been adequately represented in the training data. This implies a continued need for proofreading and correction. It is particularly important in cases where segmentation results affect diagnostic or therapeutic decisions, but it should also be considered to ensure valid conclusions when deriving image-based biomarkers for statistical evaluation within clinical and scientific studies.

Second, in order to achieve high accuracy and robustness, CNNs typically have to be trained on a large number of images for which a reliable ground truth segmentation exists. Especially when dealing with three-dimensional inputs, creating this training set completely manually can be a prohibitive effort. Once the number of annotated images is large enough to train an initial network, it is highly attractive to try out semi-automated approaches to speed up creation of a more comprehensive training set, as it is required to further increase the CNN's accuracy.

These reasons motivated us to develop a system for editing CNN-based segmentations. We focus on a specific use case, segmentation of retinal layers from Optical Coherence Tomography (OCT). OCT is a technique for obtaining three-dimensional images of the retina. It is commonly used for monitoring age related macular degeneration (AMD), which is the leading cause for irreversible vision loss in most of the developed countries [JMM08, AGS10]. The high resolution and volumetric nature of OCT make it difficult and time consuming to segment and measure AMD biomarkers, especially in epidemiological studies, which may require analysis of thousands of eyes.

The retina has a layered structure, and derivation of several AMD biomarkers involve a segmentation of retinal layers [WSB*17]. One example are drusen, which appear early on in AMD and whose size, number, and location contribute to AMD staging and tracking of disease progression. Drusen evolve as deposits of extracellular debris under the retinal pigment epithelium (RPE) layer. In a normal eye, the RPE layer should run parallel to the Bruch's membrane (BM). Therefore, being able to segment RPE and BM allows us to detect drusen as abnormal deviations between the two.

Based on a segmentation from a pretrained CNN, our proposed system introduces the following strategies for efficient semi-automated segmentation editing: First, we estimate and visualize segmentation uncertainty, in order to guide the expert towards images, and specific locations within those images, that are likely to require his or her attention. Second, we introduce two tools that allow the user to more quickly edit the CNN-based layer segmentation: The first one exploits intermediate results from the segmentation process to find the most plausible segmentation that contains

DOI: 10.2312/vcbm.20181235

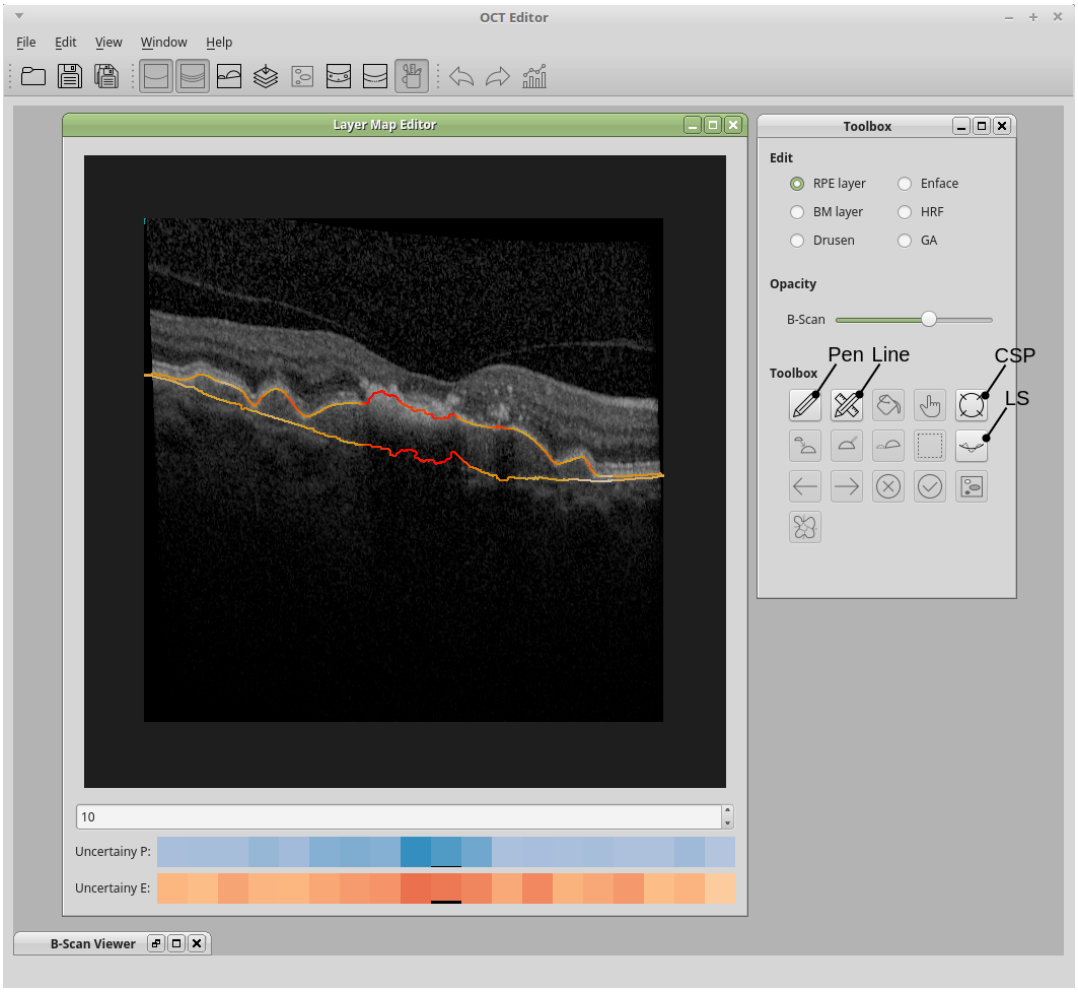

Figure 1: *Main view of our retinal layer segmentation editor. The larger subwindow overlays the CNN-based layer segmentation on the image data and color codes uncertainty both on the segmentation result itself and in a navigation bar below the viewer. Here, each cell visualizes the aggregated uncertainty of each B-scan. By clicking on a cell, the user can select between uncertainty estimates from entropy (E) or probability (P), defined in Section 4. In this figure, uncertainty P is selected for the 10th B-scan. Beside basic pen and line tools, the toolbox subwindow offers our constrained shortest path (CSP) and local smoothing (LS) tools, which are shown to greatly decrease the time required for editing the segmentation.*

specific locations indicated by the user. The second one can be used to easily smooth noisy segmentations that occur especially in regions of high uncertainty. Figure 1 shows the main view of our prototype implementation.

Throughout our discussion, we will focus on the two specific layers that are required for drusen detection, RPE and BM. However, all methods in this paper easily carry over to other retinal layers, provided that corresponding training data is available for the CNN.

We position our system with respect to existing work in Section 2, and provide a brief overview of the automated segmentation pipeline underlying our work, as well as the specific modifications that we had to make to it, in Section 3. In Section 4, we propose two measures for estimating the uncertainty of segmentation, and we introduce two special tools for segmentation correction in the next section. In Section 6, we validate our approach in two steps: First, we demonstrate a clear correlation between our uncertainty measures and a manual assessment of the segmentation quality, as well as with distance to a "ground truth" reference segmentation. Second, we asked two users to correct retinal layer segmentations in multiple test cases, and found that our novel tools allow them to achieve their task in a much shorter time.

2. Related Work

The two main aspects of our work are estimation and visualization of segmentation uncertainty on one hand, and segmentation editing on the other hand. In our discussion, we will distinguish between segmentation errors, which can be quantified by comparing a given segmentation to a reference ("ground truth"), and segmentation un-

certainty, which has to be inferred without such a reference. For example, it could be estimated from internal states of the segmentation method, or from properties of the given image data.

Visualization of segmentation errors has recently been addressed by several authors: Cárdenes et al. [CdLGC09] visualize segmentation quality based on newly derived similarity measures. Von Landesberger et al. visualize convergence of statistical shape model based segmentations to an expert segmentation [vLAA*13] and present a system for comparative evaluation across larger datasets [vLBB16]. Geurts et al. [GSK*15] visually compare segmentation quality to identify the most suitable automated algorithms. Raidou et al. [RMB*16] developed a system for visually exploring segmentation errors both in individuals and in whole cohorts.

All of the above-mentioned approaches follow a different goal than our own, which is intended for cases in which no ground truth is available, but where we would still like to guide the user towards images where corrections are most likely to be required. A previous work that follows a similar goal is the uncertainty-aware guided volume segmentation framework of Praßni et al. [PRH10]. However, it is tied to a specific algorithm, random walker segmentation, while the segmentations we would like to correct are generated via Convolutional Neural Networks and shortest path finding. Similarly, recent work by Summa et al. [STP17] on quantifying uncertainty in 2D image segmentations does not directly apply to our case, since it is specific to min-cut algorithms. Saad et al. [SHM10] take a different approach to guiding the user towards potentially erroneous segmentations, based on learning shape and appearance priors from expert segmentations. This idea does not easily carry over to our specific task of segmenting retinal layers in the presence of pathological anomalies. Yao et al. [YGLG12] propose an interactive system which we found inspiring with respect to its user guidance, but which addresses object detection rather than segmentation, and is not based on CNNs either.

Since we build on a CNN-based segmentation, we note that numerous recent works have used visualization to support better understanding of neural network architectures [TM05, EBCV09, ZF14, MV15, YCN*15, RFFT17, WSW*18, KAKC18, SGPR18], to monitor their training process [LSC*18], and to facilitate tuning of hyperparameters such as the number of neurons and layers [LSL*17, PHG*18, RG17]. However, none of them are concerned with using intermediate results to support correcting individual outcomes, which is our main focus.

We will use entropy as a building block for one of our uncertainty measures. We consider this to be a natural choice, and note that it was used previously for visualizing segmentation uncertainty by Potter et al. [PGA13]. A closely related measure was also used by Al-Taie et al. [AHL14]. We will adapt this measure to our needs, combine it with a second uncertainty measure, and validate it in the context of our application.

The need for interactive intervention in medical image segmentation has long been recognized [OS01], and methods for more efficient segmentation correction have remained an active topic [HKR*14, VFI*16]. In their overview of segmentation editing, Heckel et al. [HMTH13] distinguish between three basic types of approaches: Systems that allow for convenient parameter tuning, refinement approaches that integrate additional information about

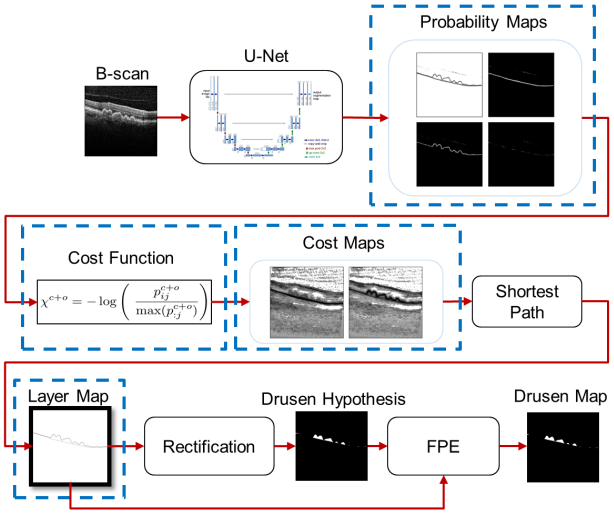

Figure 2: *In this overview of the original segmentation pipeline, the parts that had to be modified for our guided semi-automated editing framework are highlighted with dashed blue boxes.*

the desired result into an automated segmentation, and ones in which the correction is done manually, independently from the algorithm that generated the original segmentation. We provide editing tools from the second and third of these categories: Our constrained shortest path tool (Section 5.2) allows the user to specify points that should lie on the final layer segmentation. This is a classic and widely used type of interaction [NL98] which we implement for the first time in the context of our specific segmentation method. Our local smoothing tool (Section 5.3) allows the user to regularize selected parts of the given segmentation, and works independently from the underlying image and segmentation method.

Most existing software for visualizing and analyzing data from Optical Coherence Tomography is only commercially available. For instance, Stratus OCT (Carl Zeiss Meditec, Inc. Dublin, CA) can be used for measuring retinal nerve fiber layer thickness [CCAG*05]. Other existing software includes Spectralis HRA+OCT (Heidelberg Engineering, Inc., Heidelberg, Germany), Spectral OCT/SLO (Opko/OTI, Inc., Miami, FL), SOCT Copernicus (Reichert/Optopol Technology, Inc., Depew, NY), RTVue-100 (Optovue Corp., Fremont, CA), and Cirrus HD-OCT (Carl Zeiss Meditec, Inc.). They can all be used for macular thickness measuring and are compared with each other in [WSCB*09]. To the best of our knowledge, they currently do not offer CNN-based segmentations or any specific tools for guiding the user's attention to potential segmentation errors.

3. Background on Retinal Layer Segmentation

Our uncertainty estimation and one of our segmentation editing tools rely on details of the underlying method for segmenting retinal layers. To make our paper more self-contained, this section provides a brief summary of that algorithm. Its overall computational

pipeline is illustrated, and the parts that had to be modified for our editing tool are highlighted with dashed blue boxes, in Figure 2.

Retinal layer segmentation is an important and widely studied problem in medical image analysis, with a large body of literature [WSB*17]. Recently, various groups have started to leverage the power of convolutional neural networks for this application [GZWW*17, AZC*17, FCW*17, HACR*18]. We selected one of these modern approaches [GZWW*17], which has been shown to be more accurate than a traditional state-of-the-art method [CLZ*13], as the basis of our system.

Optical Coherence Tomography acquires three-dimensional images of the retina as a sequence of two-dimensional slice images, which are called B-scans. The part of the pipeline that is relevant to our system acts on these. The first step is to convert the images to a set of maps that indicate the probability of a given pixel to be part of a particular retinal layer, or to show some other structure ("background"). This is achieved with a suitably trained U-Net [RFB15], a type of convolutional neural network that is specialized for semantic image segmentation. Our editing tool uses the probability maps as the basis for uncertainty estimation.

For each layer, a cost function is used to convert its probability map into a cost map. Compared to [GZWW*17], implementing our constrained shortest path tool required a modification of that function, and updates of the cost map that reflect the user interactions. This modification is discussed in more details in Section 5.2. Dijkstra's shortest path algorithm [Ski90] is run on the (modified) cost map to extract a layer segmentation. In the fully automated approach, its result is used as is. If needed, our correction software allows the rater to modify it further using freehand manual editing, or a smoothing tool.

The rest of the pipeline is used to detect drusen based on the layer segmentations, and is less relevant to our editing system, which focuses on improving the layer segmentations.

4. Segmentation Uncertainty

When analyzing the failure cases of the automated pipeline described above, we found that they include advanced stages of AMD, and the presence of additional pathological changes. Our first goal is to guide the rater towards such problems by estimating and visualizing segmentation uncertainty.

4.1. Uncertainty Estimation

We define two different measures that estimate segmentation uncertainty, u_e and u_p. Both of them are derived from the retinal pigment epithelium (RPE) and Bruch's membrane (BM) probability maps generated by the CNN. The goal of the first measure, u_e, is to estimate how uncertain the CNN is concerning the exact localization within a given image column. This amounts to considering whether layer probabilities within a column are highly concentrated (low uncertainty) or diffuse, so that they provide some level of support for layer localization in different places (high uncertainty).

We quantify this by normalizing the probabilities $p_{ij}^{(l)}$, with $l \in \{\text{RPE}, \text{BM}\}$ being a retinal layer and (i, j) the pixel position in the probability maps, so that in each column they sum to one, i.e.,

$$q_{ij}^{(l)} = \frac{p_{ij}^{(l)}}{\sum_i p_{ij}^{(l)}}, \tag{1}$$

and then computing the information entropy of the resulting values. As there is an inverse relation between the concentration of layer probabilities in a column and the entropy value, we turn entropy into a normalized uncertainty measure $u_e^{(l,j)} \in [0,1]$ via the equation

$$u_e^{(l,j)} = 1 - G_\sigma * \exp\left(\sum_i q_{ij}^{(l)} \ln q_{ij}^{(l)}\right), \tag{2}$$

where the superscript (l, j) indicates that we consider the uncertainty in layer l and image column j. A small amount of Gaussian smoothing G_σ (in all our experiments, $\sigma = 2$) across neighboring columns is used to add robustness, so that isolated columns of high entropy do not receive an excessive weight. In our experiments Gaussian window size is set to 17 pixels, and at the boundaries, the values are reflected.

Regularization via G_σ is particularly important when aggregating per-layer/per-column uncertainties into a per-B-scan uncertainty u_e, which we achieve by taking the maximum over all $u_e^{(l,j)}$, since even increased uncertainty in a relatively small area and in one of the layers will still require the expert's attention:

$$u_e = \max_{l,j}\left(u_e^{(l,j)}\right) \tag{3}$$

The goal of our second measure, u_p, is to indicate cases in which the CNN failed to detect the presence of the respective layer with reasonable confidence. Its definition is based on probabilities $p_j^{(l)} = p_{ij}^{(l)}$ which are directly taken from the probability map of layer l, at the row index i where the shortest path traversed column j. If this probability decreases along the layer, it means that uncertainty increases:

$$u_p^{(l,j)} = 1 - G_\sigma * p_j^{(l)} \tag{4}$$

Gaussian smoothing is done for the same reason as before, and per-layer/per-column values of $u_p^{(l,j)}$ are again aggregated into a global u_p by taking the maximum over l and j.

4.2. Uncertainty Visualization

As it can be seen in the largest subwindow within Figure 1, our user interface visualizes the per-layer/per-column uncertainties on top of the segmentations. We use separate color maps for u_e (orange/red) and u_p (blue) to clarify which measure is currently shown. In both cases, we use more saturated colors to make regions of higher uncertainty stand out more from the grayscale background.

Even though we found that u_e and u_p frequently highlight the same regions as problematic, we decided to include both of them because of occasional cases in which one was more sensitive to segmentation errors than the other. Examples of this are shown in Figure 3: On the left, the segmentation failed to follow a large druse,

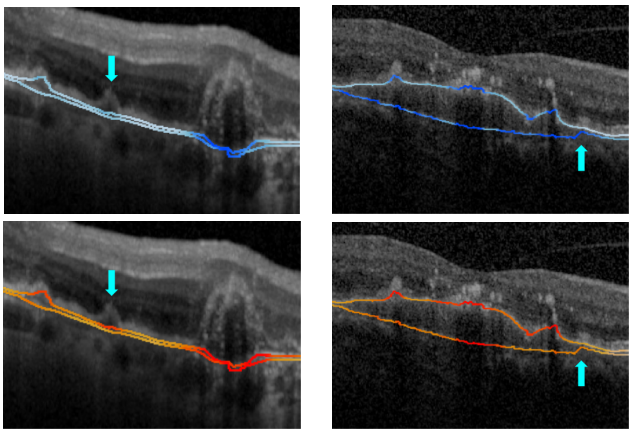

Figure 3: *Two examples that illustrate the relative strengths of our uncertainty measures u_p (top) and u_e (bottom): On the left, u_e is elevated below a missed druse that did not affect u_p. On the right, u_p indicates an implausible kink that has no clear effect on u_e.*

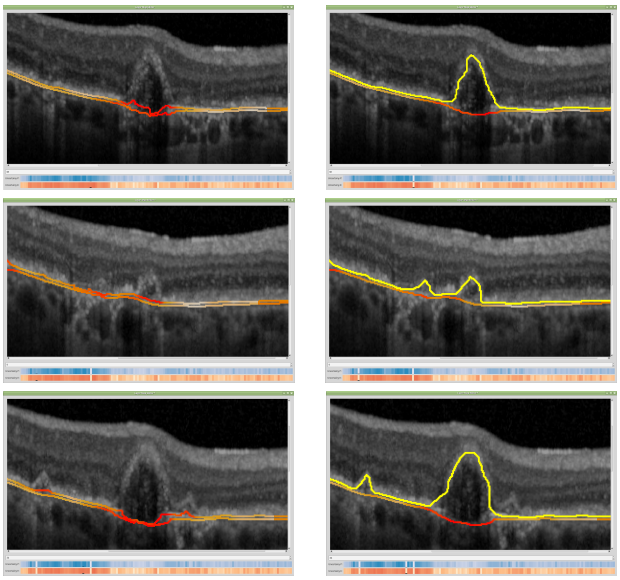

Figure 4: *Images on the left show cases in which drusen were missed in the automated RPE segmentation. Corrected results on the right are obtained with our constrained shortest path tool, by clicking once (in the top image) or twice (for the pairs of missed drusen in the other two). Shortest paths are updated to go through the selected points, while remaining smooth and on pixels that are likely to belong to the given layer.*

which was equally flagged as problematic by u_e and u_p, but also a smaller one (marked with a cyan arrow), which is indicated by an increased u_e on the RPE layer, but not reflected in u_p. On the right, the cyan arrow marks an implausible kink in the BM, on which u_p indicates increased uncertainty, but u_e is much less affected.

Below the viewer in Figure 1, there are two color coded tables. The color of the ith cell in each of them represents the aggregated uncertainty in the ith B-scan, and clicking on one of the cells shows the corresponding B-scan in the viewer. We consider these navigation tables to be even more important than the overlays on the segmentation itself, since they quickly guide the user towards the B-scans that are most likely to require attention.

5. Tools for Segmentation Editing

5.1. Basic Tools

A prerequisite for detecting and correcting segmentation errors is to overlay the layers on the original B-scan. However, in ambiguous cases, the fact that the layer segmentation occludes part of the image can impair the rater's assessment of the original data. Therefore, our interface provides a slider to modify the layers' opacity.

The layer that should be affected by our tools can be selected with radio buttons. We consider two of our editing tools to be "basic" and use them as a baseline in our evaluation: The pen allows the user to mark individual pixels as belonging to the selected layer. To speed up the correction of the segmentation, the previous layer segmentation in the columns that the pen tool is applied on, is simultaneously and automatically erased. This is based on the assumption that the layers run from left to right without looping back. It agrees with the data format in which ground truth segmentations were given to us (i.e., one coordinate per image column) and worked well in all our examples.

For parts of the layers that are rather straight, the line marker offers a more convenient basic tool. It can be used to mark every pixel along a line segment as belonging to the selected layer. The same automatic deletion of previous segmentations as in the pen tool is also included here.

5.2. Constrained Shortest Path Tool

A very common error in CNN-based segmentations of the RPE concerns cases in which they failed to follow drusen. A few examples of this can be seen in Figure 4. Since drusen are curved structures, the line marker is not very suitable for correcting these errors, and tracing them freehand with the pen tool can be tedious and time-consuming. Therefore, we were aiming to design a semi-automated tool that should be able to correct such errors by clicking on a single point along the desired (corrected) contour.

We found that, in most cases, this effect can be achieved by repeating the shortest path extraction from the original segmentation pipeline after modifying the cost maps based on the user's input. In particular, clicking on a pixel should set its cost to zero, and the cost of every other pixel in the same image column should be set to a large number. This has the effect of enforcing the new shortest path to run through the selected point.

In our initial experiments, this constrained shortest path (CSP) tool sometimes led to implausible spiky results, in which the layer took a very steep slope to interpolate the selected point. We found that this was partly caused by the exact cost function from [GZWW*17], which involved dividing layer probabilities by the maximum within the same column. Combined with the con-

straints, this sometimes made vertical moves within the cost map much cheaper than horizontal or diagonal ones. This problem was greatly reduced by replacing the maximum over the column with the maximum over the full image, i.e., our new cost function is

$$\chi^{(l)} = -\log\left(\frac{p_{ij}^{(l+o)}}{\max p_{::}^{(l+o)}}\right), \tag{5}$$

where p_{ij} is the probability at position (i, j), $\chi^{(l)}$ is the cost image for layer l, and $\max(p_{::}^{(l+o)})$ is the maximum, taken over the full B-scan, of the combined probability of l and a special class o that is used to encode layer overlaps. Class o is defined separately, since the two layers might overlap in some pathological cases.

To steer the smoothness of the CSP results, we also provide a slider that allows the user to modify a scaling factor that determines the relative cost of horizontal compared to vertical moves.

In order to allow the user to continue working on the segmentations at a later time, they can be saved (along with the modified cost maps) and reloaded later on, with all the states unchanged. To make it easy to remember which layers have already been modified, we render them in yellow. In addition, as can be seen even more clearly in Figure 5, the uncertainty values in the navigation table below the viewer are updated, i.e., are set to absolutely certain, to help the user keep track of B-scans that were already modified.

5.3. Local Smoothing Tool

Finally, we introduce a local smoothing (LS) tool that can be used in cases where the automated layer segmentation is in the correct overall location, but too "wiggly", which can happen due to noise in regions of weak contrast. This tool allows the user to select a layer and a horizontal interval on the image. It replaces the original layer segmentation in that interval with a smoothed version, which is obtained by fitting a low-degree polynomial to the pixel positions of the original segmentation. The user can control the amount of smoothing by changing the polynomial order. Figure 5 shows two examples in which Bruch's membrane was corrected with this tool.

6. Evaluation

We provide a statistical evaluation of our proposed uncertainty measures by showing that they correlate with manually derived measures of segmentation quality. We also performed experiments with test users that quantify how much time is saved by using our proposed editing tools, and by the uncertainty guidance.

6.1. Statistical Evaluation of Uncertainty Measures

In Section 4, we introduced two uncertainty measures u_e and u_p with the goal of drawing the rater's attention to B-scans that are likely to require corrections. If these measures are suitable for this purpose, they should correlate with segmentation errors as measured by more traditional means. To test this, we compared rankings of B-scans according to u_e and u_p to two other rankings: The first one is based on manually classifying segmentation quality in six different OCT scans with about 365 B-scans in total, on a scale

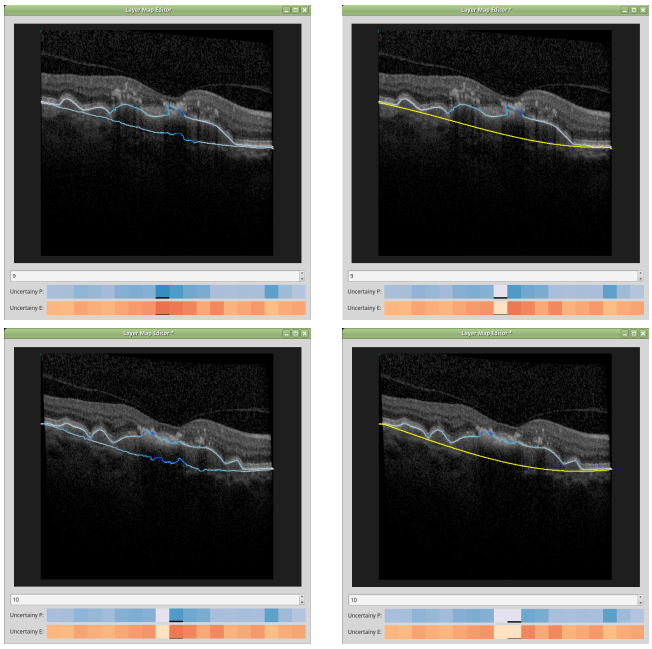

Figure 5: *Layer correction with our local smoothing tool: In the examples on the left, a lack of image contrast caused a noisy estimate of Bruch's membrane. On the right, this has been fixed using polynomial fitting. In addition, the uncertainty color codes in the navigation tables are updated.*

from 1 to 5, 1 being no further attention by user is required, and 5 indicating a strong need for correction. The ranking is done by a human rater considering the severity of segmentation failure. The second one is based on the segmentation error with respect to an expert-provided ground truth. For this, the column-wise Euclidean distance between the automated segmentation and the ground truth was taken. Since many layer segmentation errors only affect relatively small regions in the data, our error measure averages the largest 10% of per-column distances in each B-scan.

We compare the rankings based on our uncertainty measures to the manual and segmentation error based ones using Kendall's tau-b rank correlation coefficient [Ken45]. It ranges between 1 (identical rankings) and -1 (inverted rankings), and is applicable to rankings that include ties, as they will occur in our manual ranking due to the discrete nature of the underlying scale.

From the six different OCT scans, we obtain six different rank correlation coefficients, which we summarize using boxplots in Figure 6. As a reference for the magnitude of correlations that we can reasonably expect, both plots include a direct comparison between segmentation error based and manual rankings (black). The remaining comparisons are between our uncertainty measures and manual rankings (top) or segmentation errors (bottom), respectively. As uncertainty measures, we consider u_e, u_p, and the average of the two. We aggregate them over the B-scan either using the max (as suggested in Section 4.1, shown in blue), or by averaging over all image columns (red).

	Without guidance		With guidance	Without guidance		With guidance	
	Only basic	All	All	Only basic	All	All	All (repeated)
Time per B-scan (Sec)	22.1	12.8	6.9	17.6	8.5	4.1	5.2
Total time (Sec)	3206	1860	1009	2565	1227	598	760
Total actions (clicks)	576	395	208	660	327	176	196
Pen	46.4%	0.9%	0%	27.6%	6.2%	0%	0%
Line	38.8%	11.2%	10.9%	70.3%	3%	5.4%	0%
CSP	-	73%	76.2%	-	85.6%	90.5%	96.8%
LS	-	11.5%	12.4%	-	3.6%	3%	2.1%
Undo/Redo	14.8%	3.4%	0.5%	2.1%	1.6%	1.1%	1.1%
	Novice User			Experienced User			

Table 1: *Each of two test users corrected OCT scans with basic and all editing tools, and with and without uncertainty guidance. The first row shows the corresponding average time needed to correct segmentations in one B-scan. The second row shows the time spent on correcting one full OCT. The third row shows the number of total actions (clicks) per experiment, and the remaining rows reveal the percentages of using the different tools in each experiment. To confirm that remembering the required modifications is not the main factor behind the observed efficiency gain, the experienced user repeated the final part of the experiment after two months.*

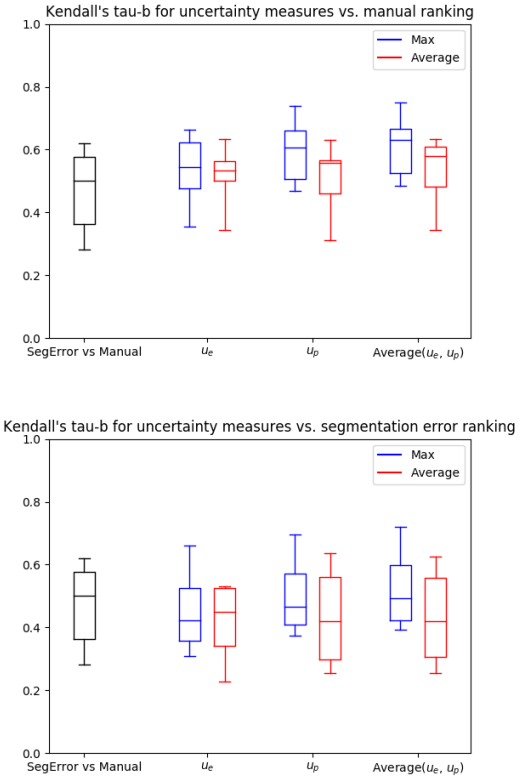

Figure 6: *Comparing our uncertainty measures to a manual rating of segmentation quality or to segmentation errors with respect to a ground truth indicates a clear correlation, which shows that they are suitable for guiding users to problematic cases.*

These results confirm our choice of max-aggregation by illustrating that it leads to slightly stronger overall correlations than simple averaging. Also, slightly better results were obtained when combining u_e and u_p than when using them individually, which confirms their complementary nature as observed qualitatively in Figure 3.

The overall correlation between our uncertainty measures and manual or error-based rankings was similarly strong as the correlation between our two references. Together with the results from the following section, this confirms their suitability for user guidance.

6.2. Quantifying the Achieved Efficiency Gain

In order to quantify how much time is saved by the uncertainty guidance and advanced editing tools, we asked two users to correct the automated segmentation of an OCT scan consisting of 145 B-scans, which we had noticed to contain serious errors. One test user was actively involved in designing the system, and will be denoted as "experienced". The other test user ("novice") was previously unfamiliar with our system. She was given about 10 minutes of training on how to use our tools, and was then given about an hour to become more familiar with them on data that also contained segmentation errors, but was not included in the experiment itself.

In order to distinguish the roles of the improved editing tools and the uncertainty guidance, each user was asked to perform the segmentation correction three times. In the first round, users were only allowed to use the basic tools (pen and line marker), and were not shown any uncertainty information. In the second round, all editing tools were provided (including CSP and LS), but still without indicating uncertainty. In the final round, users could additionally navigate through the data based on the per-B-scan uncertainty estimates, and local uncertainties were color coded on the layers, as discussed in Section 4.2.

In all cases, our software counted how frequently each of the editing tools and the undo/redo functionality was used, and we measured the time it took the users to reach a predefined level of accuracy, which was computed with respect to a ground truth reference that had been created previously by an independent expert,

using the OCT manufacturer's software. In particular, experiments were terminated when the average segmentation error fell below 5 pixels for the whole OCT scan, and there was no B-scan left that included any segmentation errors greater than 10 pixels. We used such a termination criterion to ensure that users achieve the same accuracy in each round, and do not stop prematurely. The specific tolerances were determined in a pilot experiment in which we tested how close we could reasonably expect our users to get to the ground truth without revealing it to them.

Table 1 summarizes the results of these experiments. Even though the novice user required roughly 70% more time than the experienced user, and made much more use of undo/redo, both users clearly benefitted from our novel editing tools, which reduced the time required for the corrections approximately by a factor of two. This went along with a clear reduction in the number of actions (clicks). Usage statistics suggest that this speedup was mostly enabled by the constrained shortest path tool. However, the relatively low numbers for the local smoothing tool should be seen in the light of the fact that a single application is usually enough to correct the whole BM layer. Guidance from the uncertainty visualization led to an additional reduction of the required time, by an overall factor of more than three.

To ensure that no differences in difficulty confound our findings, all three rounds of the experiments were done on the same dataset. However, this means that part of the speedup might be explained by users remembering some of the required changes. In order to account for this, we asked the experienced user to repeat the correction just once, with all advanced tools and uncertainty guidance, after two months of not viewing the data. The results of this repeated experiment are also reported in Table 1, and indeed suggest a small effect from familiarity with the data. However, the main improvement can still be attributed to the use of our framework.

We note that part of the benefit from the uncertainty visualization that was observed in our experiment could be due to the fact that our termination mechanism might have ended the experiment even before the user viewed all B-scans. When the goal is to ensure the best possible accuracy within an arbitrary amount of time, this might be considered to be unacceptable. On the other hand, if the goal is to improve the quality of a given large-scale training set (e.g., [GZWW*17] used 52377 B-scans) as much as possible within a limited time budget, we expect uncertainty guidance to improve efficiency considerably.

7. Conclusion

Even though CNNs are now widely used for image segmentation, there is still little work on efficiently editing their results. For the specific use case of retinal layer segmentation in OCT, we introduced a set of tools that make it easier for human raters to find images in which the automated segmentation may have failed, and faster to correct segmentation errors. We developed our prototype implementation using Python and Qt. We deployed it in the department of ophthalmology that our clinical co-author is affiliated with, and it is available at https://github.com/MedVisBonn/OCT-Annotation-Tool. We evaluated our system by correlating our uncertainty measures with segmentation quality and measuring the time saved by using our framework.

We expect that tools like ours will make it more efficient to generate large high-quality training sets as they are required to achieve optimum results with CNNs. In our future work, we plan to add functionality for specific AMD biomarkers including drusen, geographic atrophy, and hyperreflective foci.

References

[AGS10] ABRÀMOFF M. D., GARVIN M. K., SONKA M.: Retinal imaging and image analysis. *IEEE reviews in biomedical engineering 3* (2010), 169–208. 1

[AHL14] AL-TAIE A., HAHN H. K., LINSEN L.: Uncertainty estimation and visualization in probabilistic segmentation. *Computers & Graphics 39* (2014), 48–59. doi:10.1016/j.cag.2013.10.012. 3

[AZC*17] APOSTOLOPOULOS S., ZANET S. D., CILLER C., WOLF S., SZNITMAN R.: Pathological OCT retinal layer segmentation using branch residual u-shape networks. In *Proc. Medical Image Computing and Computer Assisted Intervention (MICCAI), Part III* (2017), pp. 294–301. doi:10.1007/978-3-319-66179-7_34. 4

[CCAG*05] CARPINETO P., CIANCAGLINI M., AHARRH-GNAMA A., CIRONE D., MASTROPASQUA L.: Custom measurement of retinal nerve fiber layer thickness using stratus oct in normal eyes. *European journal of ophthalmology 15*, 3 (2005), 360–366. 3

[CdLGC09] CÁRDENES R., DE LUIS GARCÍA R., CUADRA M. B.: A multidimensional segmentation evaluation for medical image data. *Computer Methods and Programs in Biomedicine 96*, 2 (2009), 108–124. doi:10.1016/j.cmpb.2009.04.009. 3

[CLZ*13] CHEN Q., LENG T., ZHENG L., KUTZSCHER L., MA J., DE SISTERNES L., RUBIN D. L.: Automated drusen segmentation and quantification in SD-OCT images. *Medical Image Analysis 17*, 8 (2013), 1058–1072. 4

[EBCV09] ERHAN D., BENGIO Y., COURVILLE A., VINCENT P.: *Visualizing Higher-Layer Features of a Deep Network*. Tech. Rep. 1341, University of Montreal, 2009. Presented at ICML 2009 Workshop on Learning Feature Hierarchies. 3

[FCW*17] FANG L., CUNEFARE D., WANG C., GUYMER R. H., LI S., FARSIU S.: Automatic segmentation of nine retinal layer boundaries in OCT images of non-exudative AMD patients using deep learning and graph search. *Biomedical Optics Express 8*, 5 (2017), 2732. doi:10.1364/boe.8.002732. 4

[GLGL18] GUO Y., LIU Y., GEORGIOU T., LEW M. S.: A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval 7*, 2 (2018), 87–93. doi:10.1007/s13735-017-0141-z. 1

[GSK*15] GEURTS A., SAKAS G., KUIJPER A., BECKER M., VON LANDESBERGER T.: Visual comparison of 3D medical image segmentation algorithms based on statistical shape models. In *Int'l Conf. on Digital Human Modeling (DHM)* (2015), vol. 9185 of *LNCS*, Springer, pp. 336–344. doi:10.1007/978-3-319-21070-4_34. 3

[GZWW*17] GORGI ZADEH S., WINTERGERST M. W., WIENS V., THIELE S., HOLZ F. G., FINGER R. P., SCHULTZ T.: CNNs enable accurate and fast segmentation of drusen in optical coherence tomography. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2017, pp. 65–73. 4, 5, 8

[HACR*18] HAMWOOD J., ALONSO-CANEIRO D., READ S. A., VINCENT S. J., COLLINS M. J.: Effect of patch size and network architecture on a convolutional neural network approach for automatic segmentation of OCT retinal layers. *Biomedical Optics Express 9*, 7 (2018), 3049. doi:10.1364/boe.9.003049. 4

[HKR*14] HAEHN D., KNOWLES-BARLEY S., ROBERTS M., BEYER J., KASTHURI N., LICHTMAN J. W., PFISTER H.: Design and evaluation of interactive proofreading tools for connectomics. *IEEE Trans. on Visualization and Computer Graphics 20*, 12 (2014), 2466–2475. doi:10.1109/TVCG.2014.2346371. 3

[HMTH13] HECKEL F., MOLTZ J. H., TIETJEN C., HAHN H. K.: Sketch-based editing tools for tumour segmentation in 3d medical images. *Computer Graphics Forum 32*, 8 (2013), 144–157. doi:10.1111/cgf.12193. 3

[JMM08] JAGER R. D., MIELER W. F., MILLER J. W.: Age-related macular degeneration. *New England Journal of Medicine 358*, 24 (2008), 2606–2617. 1

[KAKC18] KAHNG M., ANDREWS P. Y., KALRO A., CHAU D. H.: ActiVis: Visual exploration of industry-scale deep neural network models. *IEEE Trans. on Visualization and Computer Graphics 24*, 1 (2018), 88–97. 3

[Ken45] KENDALL M. G.: The treatment of ties in ranking problems. *Biometrika 33*, 3 (1945), 239–251. 6

[LSC*18] LIU M., SHI J., CAO K., ZHU J., LIU S.: Analyzing the training processes of deep generative models. *IEEE Trans. on Visualization and Computer Graphics 24*, 1 (2018), 77–87. 3

[LSL*17] LIU M., SHI J., LI Z., LI C., ZHU J., LIU S.: Towards better analysis of deep convolutional neural networks. *IEEE Trans. on Visualization and Computer Graphics 23*, 1 (2017), 91–100. 3

[MV15] MAHENDRAN A., VEDALDI A.: Understanding deep image representations by inverting them. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 5188–5196. 3

[NL98] NEUMANN A., LORENZ C.: Statistical shape model based segmentation of medical images. *Computerized Medical Imaging and Graphics 22*, 2 (1998), 133–143. 3

[OS01] OLABARRIAGA S. D., SMEULDERS A. W. M.: Interaction in the segmentation of medical images: A survey. *Medical Image Analysis 5*, 2 (2001), 127–142. doi:10.1016/S1361-8415(00)00041-4. 3

[PGA13] POTTER K. C., GERBER S., ANDERSON E. W.: Visualization of uncertainty without a mean. *IEEE Computer Graphics and Applications 33*, 1 (2013), 75–79. doi:10.1109/MCG.2013.14. 3

[PHG*18] PEZZOTTI N., HÖLLT T., GEMERT J. V., LELIEVELDT B. P., EISEMANN E., VILANOVA A.: DeepEyes: Progressive visual analytics for designing deep neural networks. *IEEE Trans. on Visualization and Computer Graphics 24*, 1 (2018), 98–108. 3

[PRH10] PRASSNI J.-S., ROPINSKI T., HINRICHS K.: Uncertainty-aware guided volume segmentation. *IEEE Trans. Visualization and Computer Graphics 16*, 6 (2010), 1358–1365. 3

[RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. In *Int'l Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2015), Springer, pp. 234–241. 4

[RFFT17] RAUBER P. E., FADEL S. G., FALCAO A. X., TELEA A. C.: Visualizing the hidden activity of artificial neural networks. *IEEE Trans. on Visualization and Computer Graphics 23*, 1 (2017), 101–110. 3

[RG17] ROESCH I., GÜNTHER T.: Visualization of neural network predictions for weather forecasting. In *Proc. Vision, Modeling, Visualization (VMV)* (2017), Hullin M., Klein R., Schultz T., Yao A., (Eds.), The Eurographics Association. 3

[RMB*16] RAIDOU R. G., MARCELIS F. J. J., BREEUWER M., GRÖLLER M. E., VILANOVA A., VAN DE WETERING H. M. M.: Visual analytics for the exploration and assessment of segmentation errors. In *VCBM 16: Eurographics Workshop on Visual Computing for Biology and Medicine, Bergen, Norway, September 7-9, 2016* (2016), pp. 193–202. doi:10.2312/vcbm.20161287. 3

[SGPR18] STROBELT H., GEHRMANN S., PFISTER H., RUSH A. M.: LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Trans. on Visualization and Computer Graphics 24*, 1 (2018), 667–676. 3

[SHM10] SAAD A., HAMARNEH G., MÖLLER T.: Exploration and visualization of segmentation uncertainty using shape and appearance prior information. *IEEE Trans. on Visualization and Computer Graphics 16*, 6 (2010), 1366–1375. doi:10.1109/TVCG.2010.152. 3

[Ski90] SKIENA S.: Dijkstra's algorithm. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley* (1990), 225–227. 4

[STP17] SUMMA B., TIERNY J., PASCUCCI V.: Visualizing the uncertainty of graph-based 2d segmentation with min-path stability. *Computer Graphics Forum 36*, 3 (2017), 133–143. doi:10.1111/cgf.13174. 3

[TM05] TZENG F.-Y., MA K.-L.: Opening the black box – data driven visualization of neural networks. In *Proc. IEEE Visualization* (2005), Silva C., Gröller E., Rushmeier H., (Eds.), pp. 383–390. 3

[VFI*16] VALENZUELA W., FERGUSON S. J., IGNASIAK D., DISERENS G., HÄNI L., WIEST R., VERMATHEN P., BOESCH C., REYES M.: FISICO: Fast Image SegmentatIon COrrection. *PLOS ONE 11*, 5 (2016), e0156035. doi:10.1371/journal.pone.0156035. 3

[vLAA*13] VON LANDESBERGER T., ANDRIENKO G. L., ANDRIENKO N. V., BREMM S., KIRSCHNER M., WESARG S., KUIJPER A.: Opening up the "black box" of medical image segmentation with statistical shape models. *The Visual Computer 29*, 9 (2013), 893–905. doi:10.1007/s00371-013-0852-y. 3

[vLBB16] VON LANDESBERGER T., BASGIER D., BECKER M.: Comparative local quality assessment of 3D medical image segmentations with focus on statistical shape model-based algorithms. *IEEE Trans. on Visualization and Computer Graphics 22*, 12 (2016), 2537–2549. doi:10.1109/TVCG.2015.2501813. 3

[WSB*17] WINTERGERST M. W., SCHULTZ T., BIRTEL J., SCHUSTER A. K., PFEIFFER N., SCHMITZ-VALCKENBERG S., HOLZ F. G., FINGER R. P.: Algorithms for the automated analysis of age-related macular degeneration biomarkers on optical coherence tomography: A systematic review. *Translational Vision Science & Technology 6*, 4 (2017), 10. doi:10.1167/tvst.6.4.10. 1, 4

[WSCB*09] WOLF-SCHNURRBUSCH U. E., CEKLIC L., BRINKMANN C. K., ILIEV M. E., FREY M., ROTHENBUEHLER S. P., ENZMANN V., WOLF S.: Macular thickness measurements in healthy eyes using six different optical coherence tomography instruments. *Investigative ophthalmology & visual science 50*, 7 (2009), 3432–3437. 3

[WSW*18] WONGSUPHASAWAT K., SMILKOV D., WEXLER J., WILSON J., MANE D., FRITZ D., KRISHNAN D., VIEGAS F. B., WATTENBERG M.: Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE Trans. on Visualization and Computer Graphics 24*, 1 (2018), 1–12. 3

[YCN*15] YOSINSKI J., CLUNE J., NGUYEN A. M., FUCHS T. J., LIPSON H.: Understanding neural networks through deep visualization. *CoRR abs/1506.06579* (2015). Presented at ICML 2015 Deep Learning Workshop. arXiv:1506.06579. 3

[YGLG12] YAO A., GALL J., LEISTNER C., GOOL L. J. V.: Interactive object detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2012), pp. 3242–3249. 3

[ZF14] ZEILER M. D., FERGUS R.: Visualizing and understanding convolutional networks. In *Proc. European Conf. on Computer Vision (ECCV)* (2014), vol. 8689 of *LNCS*, Springer, pp. 818–833. 3