

AUTOMATIC SEGMENTATION AND AREA ESTIMATION OF LASER ABLATION CONTOURS

Jan Hering, Jan Kybic

Department of Cybernetics, Faculty of Electrical Engineering,
Czech Technical University in Prague, Czech Republic

ABSTRACT

We evaluate the feasibility of end-to-end trained convolutional neural networks (CNN) with U-Net architecture in solving the task of automated segmentation for in-situ characterization and optimization for focused XUB/X-ray laser beam.

1. INTRODUCTION

2. METHODS

2.1. [Internal] Software and data

The source code is located in a FEE-GitLab repository¹. The code is organized in several directories, basic usage examples are provided in the README.md document in the root directory of the repository.

Original input data are stored on datagrid in the directory Medical/chalupsky/#5/Measurement. The experiments were performed on the cluster-machine zorn. The module m1 PyTorch/1.1.0-fosscuda-2019a-Python-3.7.2 provides a recent version of the GPU-capable deep learning library PyTorch, additional module requirements are specified in the requirements.txt file in the directory. Use the file to set-up a virtual environment.

2.2. Data characteristics

Our collection contains 640 images organized in 8 groups – 8 different attenuation settings with 80 images each. After exclusion of partially annotated images – those, which contain multiple spots but only one of them is annotated – we have 630 images for training/testing. Representative cases of the different image types present in the data collection are displayed on Figure 1. We can observe, that the background color changes from *blue* (Figure 1-A) to *ochre* (Figure 1-B) – this actually happen in each group – and in addition the background area may contain some heavy (Figure 1-C) or moderate (Figure 1-D) background clutter artifacts.

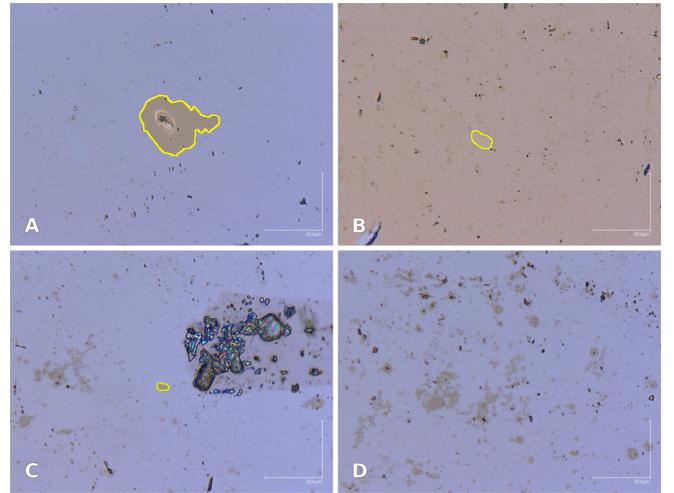


Fig. 1: Sample images from the data collection with ground-truth segmentation boundary over-painted in magenta. (A) A clear image with large attenuation contour, (B) image with small contour and different color characteristics, (C) image with heavy artifact and small contour and (D) an image without attenuation contour and moderate artifacts.

2.3. Network

The classification network $\mathbf{U} = \mathbf{U}(l_U, k_U)$ has a U-net architecture [1] and is parametrized by the number of descent levels l_U and the numbers of convolution kernels in the first dimension k_U . Higher values of l_U or k_U should lead to better segmentation results (up to a certain limit) but will increase the total number of trainable network parameters and thus the memory consumption during training. Network \mathbf{U} is trained to minimize the binary Cross-Entropy loss \mathcal{L} using the *Stochastic Gradient Descent* (SGD) optimizer with *Nesterov*'s momentum. We use a dynamic schedule for the learning rate γ_U , which is decreased by factor 0.1 iff the validation loss reaches a plateau – i.e. the validation loss stops improving for N_{lr} consecutive epochs.

The network \mathbf{U} accepts images of arbitrary size, we will use rectangular images of size $N \times N$. Larger input size comes with higher memory demands and limits together with

¹<https://gitlab.fel.cvut.cz/biomedical-imaging-algorithms/laserbeam>

the network’s size the number of images that can be processed together within a batch. Since the target spots are typically located in the central region of the input image, we will crop the input image to its central region of the desired network input’s size $N \times N$. In total, we want to find the best *trade-off* between the hyper-parameters N, B_n, l_U, k_U and the network’s learning performance.

3. EXPERIMENTS AND RESULTS

Evaluation metrics

Given a trained model \mathbf{U} and a testing image X_t , we compute the prediction map $\mathbf{U}(X_t)$ and binarize it at threshold $\vartheta \in (0, 1)$. The map $\mathbf{U}(X_t)$ is of the same size as X_t and contains the probability $P(x = 1)$ for each pixel x of X_t . From the resulting segmentation $\hat{Y}_{t,\vartheta} = \mathbf{U}(X_t) > \vartheta$ we compute the Dice score

$$D(X_t, Y_t^*, \vartheta) = \frac{2 \cdot |Y_t^* \cap \hat{Y}_{t,\vartheta}|}{|Y_t^*| + |\hat{Y}_{t,\vartheta}|} \quad (1)$$

and the relative area error

$$\text{RE}_{\text{Area}}(X_t, Y_t^*, \vartheta) = \frac{|Y_t^*| - |\hat{Y}_{t,\vartheta}|}{|Y_t^*|} \quad (2)$$

When plotting Dice score and the area error together, we transform the relative area error to the error score

$$1 - |\text{RE}_{\text{Area}}(X_t, Y_t, \vartheta)| \quad (3)$$

By this transform, both Dice end error scores tend towards 1.0 with improving predictions.

3.1. Learning directly on image data

We train a model $\mathbf{U} = U(l_U = 4, k_U = 32, B_n = 2)$ for 40 epochs while using the splits of 0.7, 0.1, 0.2 to construct the training, validation and testing sets respectively from all available data. We start with learning rate 0.001 and use the plateau-based learning rate adaptation scheme. From each input image, a training image X is constructed by cropping the central region of size 1024×1024 (in this case, the batch size is $B_n = 2$ when computing on a 12 GB GPU card). During training, data is augmented by random flips and random variations in hue and saturation. We present the distribution of the relative error in the testing set in Figure 4(a).

3.2. Histogram equalization

We use the same parameters as in Experiment 3.1 but limit the color variance in the training images with histogram equalization prior to cropping. The equalization is done by applying a non-linear mapping to the input image, in order to create a uniform distribution of gray-scale values in the equalized image [2]. We again show the distribution of the relative error in the testing set (see Figure 4(b)).

To have the full ablation contour covered in all testing images, we consider the central image region of size 1200×1200 px, and subdivide it to a matrix of 2×2 patches of size 640×640 px (network’s input size) and 80 px overlap band. We stitch the individual patch predictions together, and take the average value in overlapping regions.

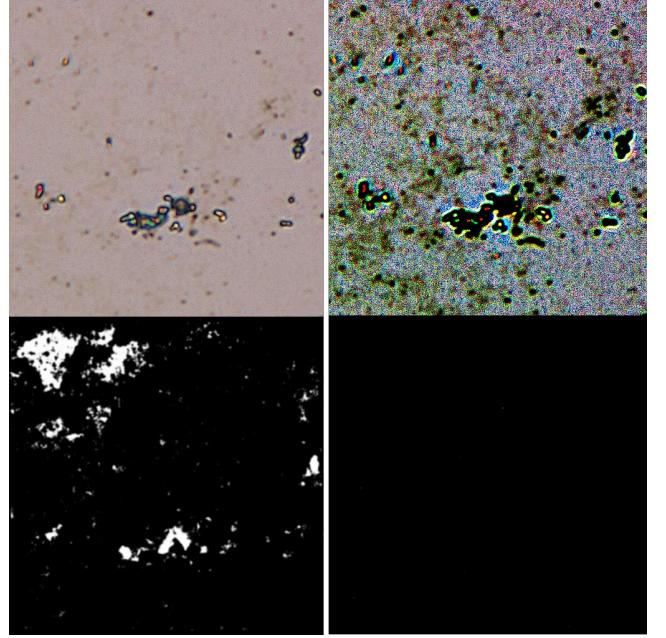


Fig. 2: *Upper row:* Example image from scan F-scan 08 shown in stored (left) and histogram-equalized (right) fashion. *Lower row:* Corresponding prediction maps after 40 epochs of training.

3.3. Adaptive normalization

Figure 2 illustrates some of the improvements achieved by the model on histogram-equalized images, but it also shows how the noise part of the image gets magnified. We reduce this effect, by replacing the previous equalization with the *Contrast Limited Adaptive Histogram Equalization* (CLAHE) [3] method. A comparison of HE and CLAHE are shown in Figure 3. So in total, we apply

- CLAHE
- crop to central region of size 640×640 pixels

during preprocessing of the input data and we generate new training images by applying following geometrical and color-based augmentations to existing images:

- 90deg rotations
- horizontal flip, vertical flip
- random distortion in hue-saturation

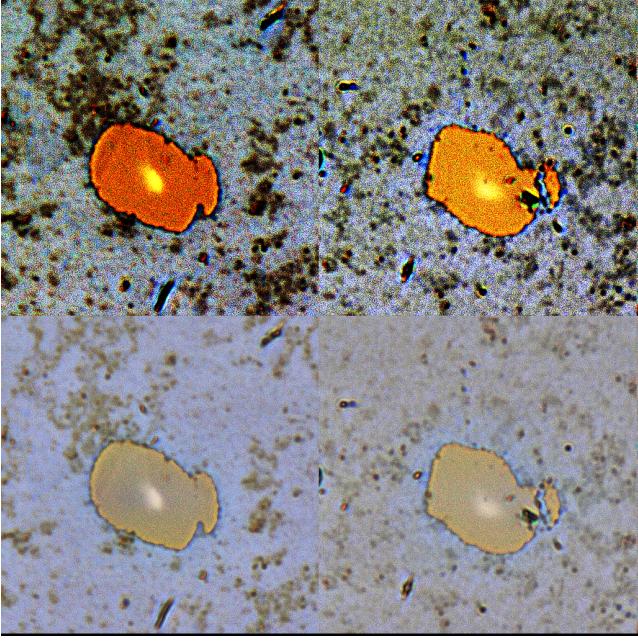


Fig. 3: Upper row: Two example images normalized with standard histogram equalization, Bottom row: same images normalized with CLAHE with clip limit of 2.0.

- random distortion of the RGB-values

The model achieved a median Dice score above 0.98 in the first 6 scan groups, the first outlier appeared for the scan group *FScan 5*. The respective images and prediction for the outlier cases are shown in Figure 8. Examples of four successful segmentation results are shown in Figure 9. There are almost no area underestimates and only few outliers when compared to the previous two experiments (direct image data and histogram equalization). All outliers are visible in Figure 7, where the Dice and error scores statistics are displayed separately for each scan group. Corresponding numerical values of the statistics are displayed in Table 1.

On Figure 6, we present the relative error in the validation set, in dependency to the binarization threshold ϑ which is used to set the *optimal* ϑ^* for the testing set. The relative error distribution in the testing set (at threshold ϑ^*) is shown on Figure 5. We see, that the amount of zero-error cases increases from *direct application* to *histogram equalization* with the overall best results for *adaptive normalization*.

4. DISCUSSION

Training a 4 level U-net with 32 kernels in the first convolutional layer provides a model, that achieves reasonable precision in spot detection and area estimation.

The classifier detects the spots correctly in most images, even when small artifacts are present around or within the spot boundary (see Figure 9, but typically fails at images with

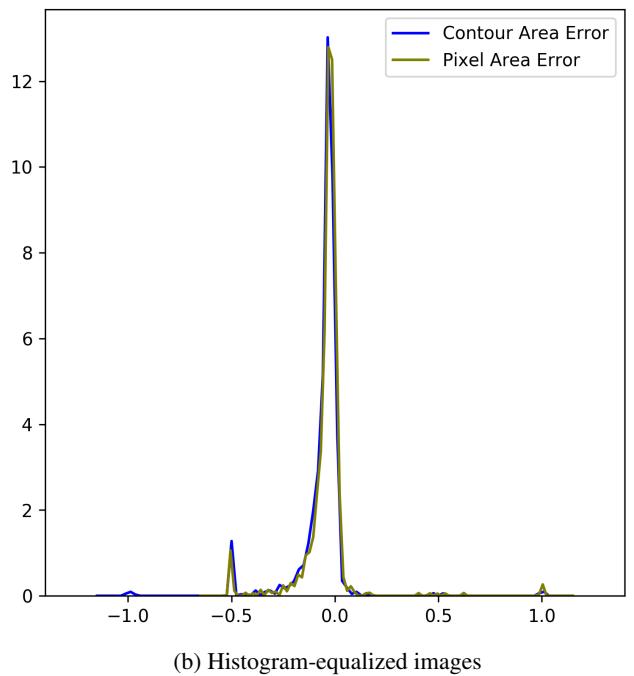
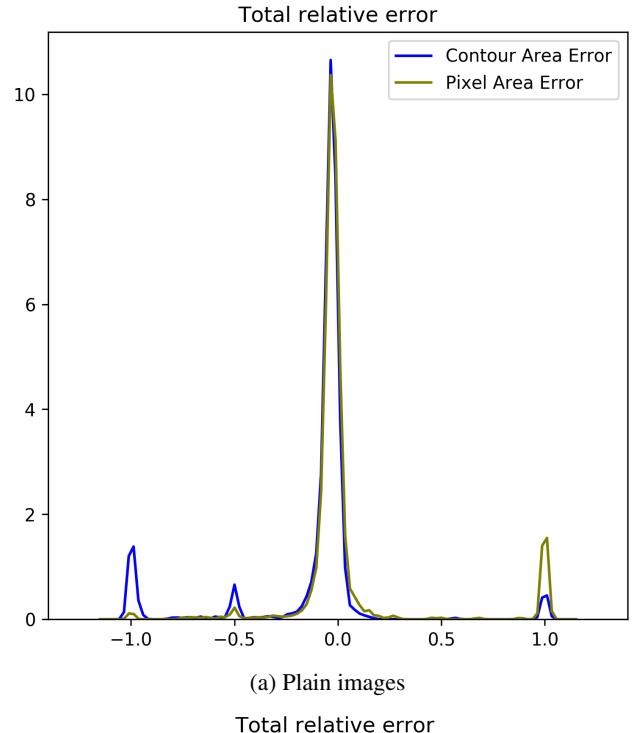


Fig. 4: Distribution of area errors within the testing set. Shown for segmentation estimated from contour fitted to the prediction (blue) and pixel-wise thresholding (green). Bin width for density estimation set to 0.05. Negative values mean area under-estimation and positive values over-estimation. Relative error of -1.0 means the predicted segmentation missed the target fully, value 1.0 means a factor at least two between predicted area and the ground truth.

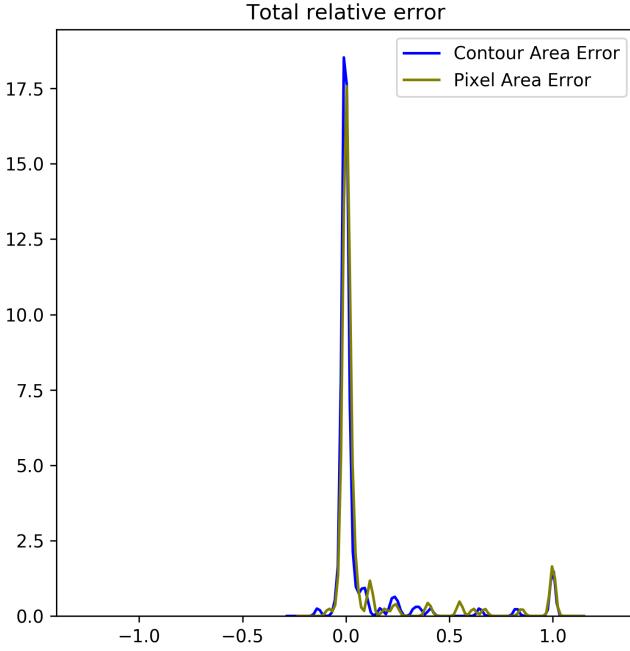


Fig. 5: Distribution of area errors within the testing set. Shown for segmentation estimated from contour fitted to the prediction (blue) and pixel-wise thresholding (green). Bin width for density estimation set to 0.05. Negative values mean area under-estimation and positive values over-estimation.

high background clutter artifacts on the other (see Figure 8). This happens because there are very few training examples containing the same type of artifact.

4.1. Further steps

There are many possible ways to follow in order to resolve the segmentation errors that occurred in the reported experiments

- **prediction post-processing** – we can observe, that the true spot has a smooth boundary and rather compact shape, whereas the outlier detection typically come with complicated shapes.
- **more data** – increase the size of the training set, primary by acquiring more, optionally with more augmentations
- **improve data augmentations** – evaluate more settings of free parameters in color augmentation (how much the hue-saturation or brightness channels are distorted), the same for the clip-limit parameter of CLAHE
- **multi-scale multi-resolution processing**, selection of approaches:
 1. a two step solution with a *localization* network, that approximately detects the spots in down-

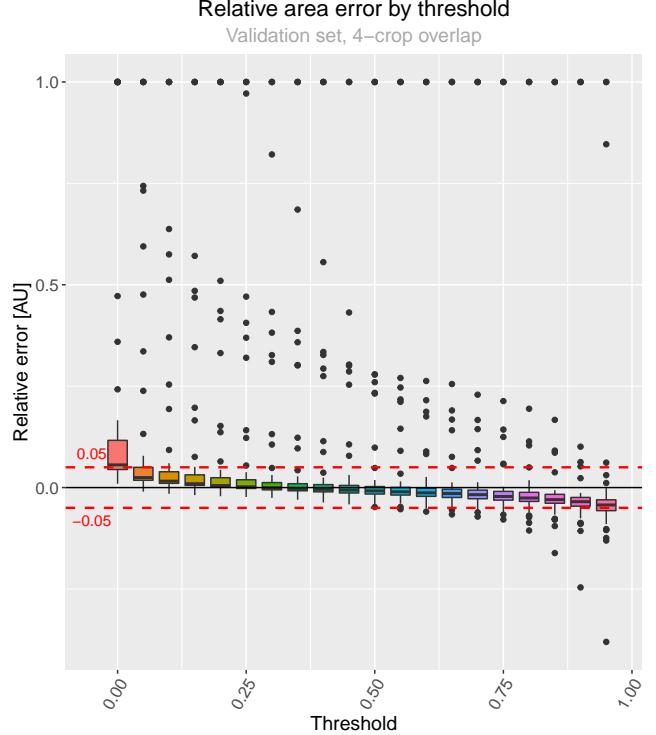


Fig. 6: Relative area error scores at different thresholds. Each column summarizes the relative error score Eq. (3) on the whole validation set. The value range of (-0.05, 0.05) is emphasized by dashed red lines.

scaled version of the input images followed by the classification with \mathbf{U} as shown

2. construct a single, *multi-scale/multi-resolution* U-net model \mathbf{U}' that explicitly processes the input images at different zooming levels, either implicitly by changing the architecture like in MultiResUNet [4], or
3. explicitly by having separate encoder paths (downward direction) for different resolutions and merging these within one common decoder path (the upward direction) [5], or solutions like U-Net++ [6] with dense skip connections, multiple decoder paths)

5. REFERENCES

- [1] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *MICCAI*, 2015, pp. 234–241.
- [2] J. C. Russ, *The image processing handbook*. CRC press, 2016.
- [3] A. M. Reza, “Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image

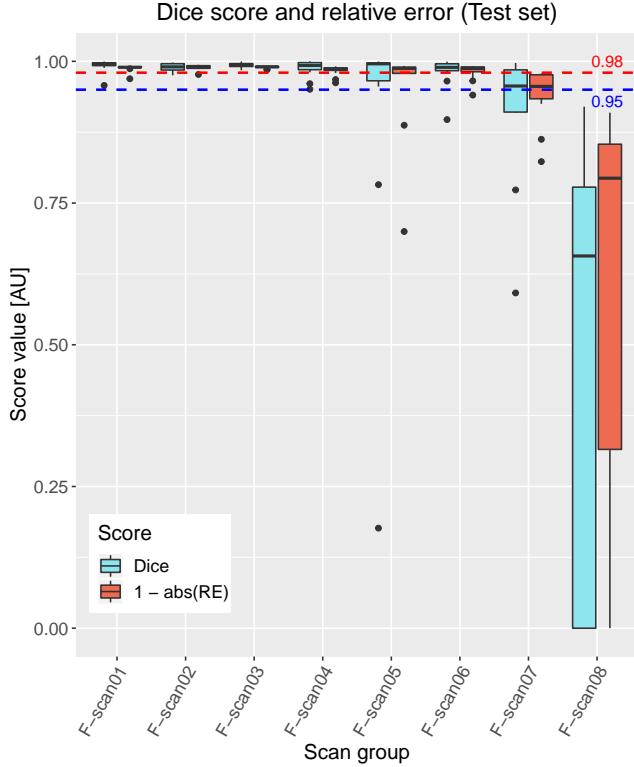


Fig. 7: Dice score and area error on the test set by scan group at binarization threshold $\vartheta^* = 0.45$. The dashed lines mark the score values of 0.98 (red) and 0.95 (blue). We plot the relative error as $1 - |\text{RE}_{\text{Area}}|$ to have the same value range as the Dice score.

enhancement,” *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 38, no. 1, pp. 35–44, 2004.

- [4] N. Ibtehaz and M. S. Rahman, “Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation,” *Neural Networks*, vol. 121, pp. 74–87, 2020.
- [5] L. Ding, K. Zhao, X. Zhang, X. Wang, and J. Zhang, “A lightweight u-net architecture multi-scale convolutional network for pediatric hand bone segmentation in x-ray image,” *IEEE Access*, 2019.
- [6] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 3–11.

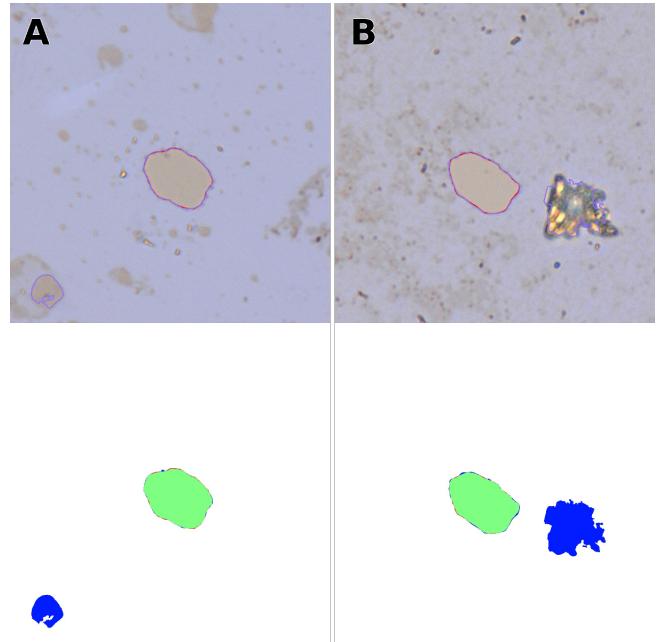


Fig. 8: Visualization of the two outliers in *FScan-5* group

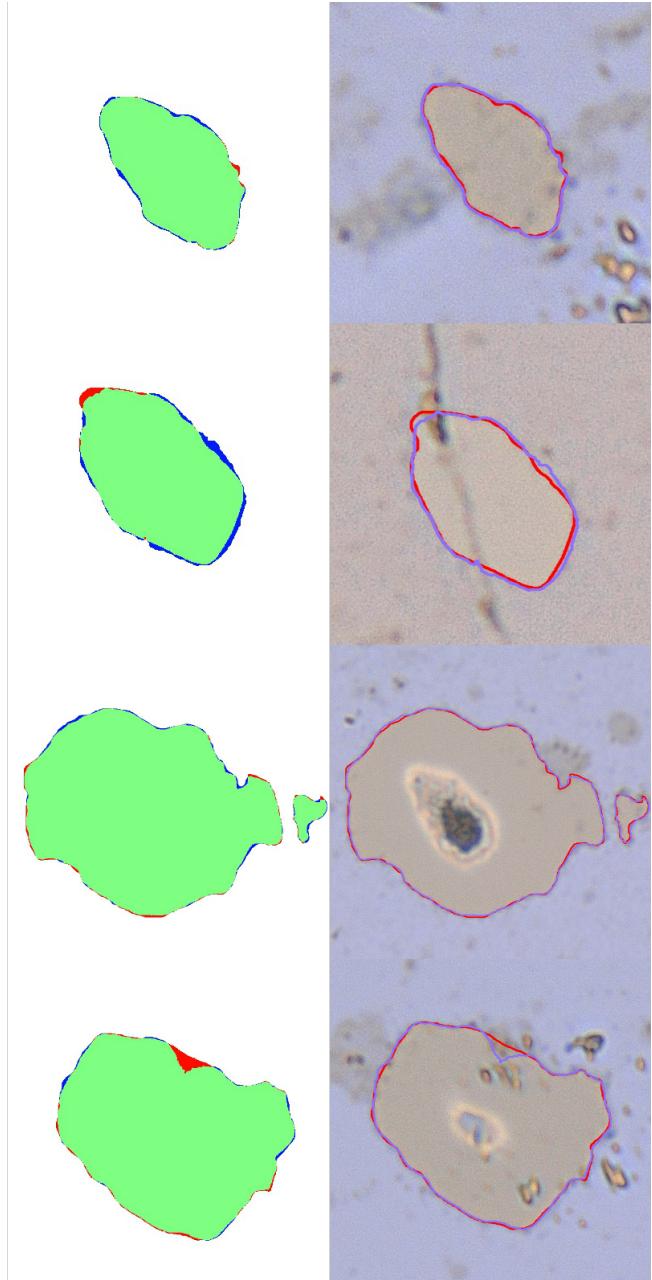


Fig. 9: Example contours. Right column shows the ground-truth contour in red and the predicted contour in magenta. Left the categorization of pixels in *TP* (green), *FN* (blue) and *FP* (red).

Table 1: Dice score and relative area error at ϑ^* . – (JH) Two of the outlier cases in scan group F-scan 8 occurred for images without visible contour – ground-truth area is 1 px, predicted area is 3 px.

Var	Group	n	Min	Median	Mean	Max	SD
Relative error	F-scan01	11	-0.011	-0.002	0.001	0.043	0.015
	F-scan02	17	-0.025	-0.010	-0.010	0.008	0.008
	F-scan03	22	-0.016	-0.005	-0.004	0.010	0.007
	F-scan04	20	-0.049	0.001	-0.002	0.039	0.017
	F-scan05	11	-0.044	-0.002	0.088	0.823	0.254
	F-scan06	18	-0.018	-0.002	0.004	0.103	0.028
	F-scan07	12	0.003	0.044	0.086	0.409	0.119
	F-scan08	14	-0.136	0.502	0.567	1.000	0.423
all		125	-0.136	-0.001	0.078	1.000	0.239
Dice score	F-scan01	11	0.969	0.990	0.988	0.992	0.006
	F-scan02	17	0.977	0.990	0.989	0.994	0.004
	F-scan03	22	0.985	0.990	0.990	0.993	0.002
	F-scan04	20	0.962	0.986	0.984	0.991	0.007
	F-scan05	11	0.700	0.988	0.952	0.992	0.089
	F-scan06	18	0.940	0.987	0.983	0.992	0.013
	F-scan07	12	0.823	0.956	0.941	0.978	0.050
	F-scan08	14	0.000	0.629	0.491	0.909	0.404
all		125	0.000	0.988	0.924	0.994	0.205